

# Constraint LTL Satisfiability Checking without Automata

MARCELLO M. BERSANI and ACHILLE FRIGERI and ANGELO MORZENTI and  
MATTEO PRADELLA and MATTEO ROSSI and PIERLUIGI SAN PIETRO,

Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy

This paper introduces a novel technique to decide the satisfiability of formulae written in the language of Linear Temporal Logic with Both future and past operators and atomic formulae belonging to constraint system  $\mathcal{D}$  (CLTLB( $\mathcal{D}$ ) for short). The technique is based on the concept of *bounded satisfiability*, and hinges on an encoding of CLTLB( $\mathcal{D}$ ) formulae into QF-EUD, the theory of quantifier-free equality and uninterpreted functions combined with  $\mathcal{D}$ . Similarly to standard LTL, where bounded model-checking and SAT-solvers can be used as an alternative to automata-theoretic approaches to model-checking, our approach allows users to solve the satisfiability problem for CLTLB( $\mathcal{D}$ ) formulae through SMT-solving techniques, rather than by checking the emptiness of the language of a suitable automaton  $\mathcal{A}_\phi$ . The technique is effective, and it has been implemented in our Zot formal verification tool.

Categories and Subject Descriptors: F.4.1 [Mathematical Logic and Formal Languages]: Mathematical Logic—Temporal logic; F.4.1 [Mathematical Logic and Formal Languages]: Formal Languages—Decision problems

General Terms: Verification, Bounded Satisfiability Checking, Completeness

## ACM Reference Format:

ACM Trans. Comput. Logic V, N, Article A (January YYYY), 39 pages.

DOI = 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

## 1. INTRODUCTION

Finite-state system verification has attained great successes, both using automata-based and logic-based techniques. Examples of the former are the so-called explicit-state model checkers [Holzmann 1997] and symbolic model checkers [Clarke et al. 1996]. However, some of the best results have been obtained by logic-based techniques, such as Bounded Model Checking (BMC) [Biere et al. 1999], a fully automated (although potentially incomplete) procedure. In BMC, a finite-state machine  $A$  (typically, a version of Büchi Automata) and a desired property  $P$  expressed in Propositional Linear Temporal Logic (PLTL) are translated into a Boolean formula  $\phi$  to be fed to a SAT solver. The translation is made finite by bounding the number of time instants. However, infinite behaviors, which are crucial in proving, e.g., liveness properties, are also considered by using the well-known property that a Büchi Automaton accepts an infinite behavior if, and only if, it accepts an infinite periodic behavior. Hence, chosen a bound  $k > 0$ , a Boolean formula  $\phi_k$  is built, such that  $\phi_k$  is satisfiable if and only if there exists an infinite periodic behavior of the form  $\alpha\beta^\omega$ , with  $|\alpha\beta| \leq k$ , that is compatible with system  $A$  while violating property  $P$ . This procedure allows counterexample detection, but it is not complete, since the violations of property  $P$  requiring “longer” behaviors, i.e., of the form  $\alpha\beta^\omega$  with  $|\alpha\beta| > k$ , are not detected. However, in many practical cases it is possible to find bounds large enough for representing counterexamples, but small enough so that the SAT solver can actually find them in a reasonable time.

Clearly, the BMC procedure can be used to check satisfiability of a PLTL formula, without considering a finite state system  $A$ . This has practical applications, since a PLTL formula can represent

This research was partially supported by Programme IDEAS-ERC and Project 227977-SMScom.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© YYYY ACM 1529-3785/YYYY/01-ARTA \$10.00

DOI 10.1145/0000000.0000000 <http://doi.acm.org/10.1145/0000000.0000000>

both the system and the property to be checked (see, e.g., [Pradella et al. 2012], where the translation into Boolean formulae is made more specific for dealing with satisfiability checking and metric temporal operators). We call this case *Bounded Satisfiability Checking* (BSC), which consists in solving a so-called Bounded Satisfiability Problem: Given a PLTL formula  $P$ , and chosen a bound  $k > 0$ , define a Boolean formula  $\phi_k$  such that  $\phi_k$  is satisfiable if, and only if, there exists an infinite periodic behavior of the form  $\alpha\beta^\omega$ , with  $|\alpha\beta| \leq k$ , that satisfies  $P$ .

More recently, great attention has been given to the automated verification of *infinite*-state systems. In particular, many extensions of temporal logic and automata have been proposed, typically by adding integer variables and arithmetic constraints. For instance, PLTL has been extended to allow formulae with various kinds of arithmetic constraints [Comon and Cortier 2000; Demri and D'souza 2002]. This has lead to the study of CLTL( $\mathcal{D}$ ), a general framework extending the future-fragment of PLTL by allowing arithmetic constraints belonging to a generic constraint system  $\mathcal{D}$ . The resulting logics are expressive and well-suited to define infinite-state systems and their properties, but, even for the bounded case, their satisfiability is typically undecidable [Demri and Gascon 2006], since they can simulate general two-counter machines when  $\mathcal{D}$  is powerful enough (e.g., Difference Logic).

However, there are some decidability results, which allow in principle for some kind of automatic verification. Most notably, satisfiability of CLTL( $\mathcal{D}$ ) is decidable (in PSPACE) when  $\mathcal{D}$  is the class of Integer Periodic Constraints (IPC\*) [Demri and Gascon 2007], or when it is the structure  $(D, <, =)$  with  $D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$  [Demri and D'Souza 2007]. In these cases, decidability is shown by using an automata-based approach similar to the standard case for LTL, by reducing satisfiability checking to emptiness verification of Büchi automata. Given a CLTL( $\mathcal{D}$ ) formula  $\phi$ , with  $\mathcal{D}$  as in the above cases, it is in fact possible to define an automaton  $\mathcal{A}_\phi$  such that  $\phi$  is satisfiable if, and only if, the language recognized by  $\mathcal{A}_\phi$  is not empty.

These results, although of great theoretical interest, are not well suited for a direct implementation, since the involved constructions are very inefficient.

In this paper, we extend the above results to a more general logic, called CLTLB( $\mathcal{D}$ ), which is an extension of PLTLB (PLTL with Both future and past operators) with arithmetic constraints in constraint system  $\mathcal{D}$ , and consider a procedure for satisfiability verification that does not rely on automata constructions. This procedure is implemented in the Zot toolkit<sup>1</sup>, verified by standard SMT-solvers, such as z3 [Microsoft Research 2009].

The idea of the procedure is to verify satisfiability by checking a finite number of  $k$ -satisfiability problems. Informally,  $k$ -satisfiability amounts to looking for ultimately periodic *symbolic* models of the form  $\alpha\beta^\omega$ , i.e., such that prefix  $\alpha\beta$  of length  $k$  admits a bounded arithmetic model (up to instant  $k$ ). Although the  $k$ -bounded problem is defined with respect to a bounded arithmetical model, it provides a finite representation of infinite symbolic models by means of ultimately periodic words. When CLTLB( $\mathcal{D}$ ) has the property that its ultimately periodic symbolic models, of the form  $\alpha\beta^\omega$ , always admit an arithmetic model, then the  $k$ -satisfiability problem can be reduced to satisfiability of QF-EUD (the theory of quantifier-free equality and uninterpreted functions combined with  $\mathcal{D}$ ). In this case,  $k$ -satisfiability is equivalent to satisfiability over infinite models.

Symmetrically to standard LTL, where bounded model-checking and SAT-solvers can be used as an alternative to automata-theoretic approaches to model-checking, reducing satisfiability to  $k$ -satisfiability allows SMT-solvers to be used in solving satisfiability for CLTLB( $\mathcal{D}$ ) formulae, instead of checking emptiness of a Büchi automaton. Moreover, when the length of all prefixes  $\alpha\beta$  to be tested is bounded by some finite  $K$ , then the number of bounded problems to be solved is also bounded. Therefore, we also prove that  $k$ -satisfiability is *complete* with respect to the satisfiability problem, i.e., by checking at most  $K$  bounded problems satisfiability of CLTLB( $\mathcal{D}$ ) formulae can always be answered.

The paper is organized as follows. Section 2 describes CLTL( $\mathcal{D}$ ) and the richer language CLTLB( $\mathcal{D}$ ), and their main known decidability techniques and results. Section 3 defines the  $k$ -

<sup>1</sup><http://zot.googlecode.com>

satisfiability problem and the actual Boolean encoding of a CLTLB( $\mathcal{D}$ ) formula is presented in Section 4. Section 5 shows the correctness of the encoding. Section 6 proves that, provided that  $\mathcal{D}$  satisfies suitable conditions, the procedure of Section 4 always terminates, due to the existence of a completeness threshold. Section 7 describes relevant related works. Finally, Section 8 concludes the paper highlighting some possible applications of the decision procedure for CLTLB( $\mathcal{D}$ ) implemented in the Zot automated verification tool.

## 2. PRELIMINARIES

This section presents an extension to Kamp's [Kamp 1968] PLTLB, by allowing formulae over a constraint system. As suggested in [Comon and Cortier 2000], and unlike the approach of [Demri 2004], the propositional variables of this logic are Boolean terms or atomic arithmetic constraints.

### 2.1. Language of constraints

Let  $V$  be a finite set of variables; a *constraint system* is a pair  $\mathcal{D} = (D, \mathcal{R})$  where  $D$  is a specific domain of interpretation for variables and constants and  $\mathcal{R}$  is a family of relations on  $D$  that is closed under complement. An *atomic  $\mathcal{D}$ -constraint* is a term of the form  $R(x_1, \dots, x_n)$ , where  $R$  is an  $n$ -ary relation of  $\mathcal{R}$  on domain  $D$  and  $x_1, \dots, x_n$  are variables. A  $\mathcal{D}$ -valuation is a mapping  $v : V \rightarrow D$ , i.e., an assignment of a value in  $D$  to each variable. A constraint is *satisfied* by a  $\mathcal{D}$ -valuation  $v$ , written  $v \models_{\mathcal{D}} R(x_1, \dots, x_n)$ , if  $(v(x_1), \dots, v(x_n)) \in R$ .

In Section 5.1 we consider  $\mathcal{D}$  to be Integer Periodic Constraints (IPC\*) or its fragments (e.g.,  $(\mathbb{Z}, <, =)$  or  $(\mathbb{N}, <, =)$ ) and  $(D, <, =)$  when  $<$  is a dense order without endpoints, e.g.,  $D = \mathbb{R}, \mathbb{Q}$ . The language IPC\* is defined by the following grammar, where  $\xi$  is the axiom:

$$\begin{aligned} \xi &:= \theta \mid x < y \mid \xi \wedge \xi \mid \neg \xi \\ \theta &:= x \equiv_c d \mid x \equiv_c y + d \mid x = y \mid x < d \mid x = d \mid \theta \wedge \theta \mid \neg \theta \end{aligned}$$

where  $x, y \in V$ ,  $c \in \mathbb{N}^+$  and  $d \in \mathbb{Z}$ . The first definition of IPC\* can be found in [Demri and Gascon 2005]; it is different from ours since it allows existentially quantified formulae (i.e.,  $\theta := \exists x \theta$ ) to be part of the language. However, since IPC\* is a fragment of Presburger arithmetic, it has the same expressivity as the above quantifier-free version (but with an exponential blow-up to remove quantifiers). Its restriction IPC<sup>++</sup> is the language defined by considering  $\theta$ , rather than  $\xi$ , as the axiom in the above grammar.

Given a valuation  $v$ , the satisfaction relation  $\models_{\mathcal{D}}$  is defined:

- $v \models_{\mathcal{D}} x \sim y$  iff  $v(x) \sim v(y)$ ;
- $v \models_{\mathcal{D}} x \sim d$  iff  $v(x) \sim d$ ;
- $v \models_{\mathcal{D}} x \equiv_c d$  iff  $v(x) - d = kc$  for some  $k \in \mathbb{Z}$ ;
- $v \models_{\mathcal{D}} x \equiv_c y + d$  iff  $v(x) - v(y) - d = kc$  for some  $k \in \mathbb{Z}$ ;
- $v \models_{\mathcal{D}} \xi_1 \wedge \xi_2$  iff  $v \models_{\mathcal{D}} \xi_1$  and  $v \models_{\mathcal{D}} \xi_2$ ;
- $v \models_{\mathcal{D}} \neg \xi$  iff  $v \not\models_{\mathcal{D}} \xi$ ;

where  $\sim$  is either  $=$  or  $<$ . A constraint is *satisfiable* if there is a valuation  $v$  such that  $v \models_{\mathcal{D}} \xi$ . Given a set of IPC\* constraints  $C$ , we write  $v \models_{\mathcal{D}} C$  when  $v \models_{\mathcal{D}} \xi$  for every  $\xi \in C$ .

### 2.2. Syntax of CLTLB

Let  $\mathcal{D} = (D, \mathcal{R})$  be a constraint system. CLTLB( $\mathcal{D}$ ) is defined as an extension of PLTLB, where atomic formulae are relations from  $\mathcal{R}$  over arithmetic temporal terms defined in  $\mathcal{D}$ . The resulting logic is actually equivalent to the quantifier-free fragment of first-order LTL over signature  $\mathcal{R}$ . Let  $x$  be a variable, *arithmetic temporal terms* (a.t.t.) are defined as:

$$\alpha := c \mid x \mid X\alpha \mid Y\alpha.$$

where  $c$  is a constant in  $D$  and  $x$  denotes variables over  $D$ . The syntax of (well formed) formulae of CLTLB( $\mathcal{D}$ ) is recursively defined as follows:

$$\phi := R(\alpha_1, \dots, \alpha_n) \mid \phi \wedge \phi \mid \neg\phi \mid \mathbf{X}\phi \mid \mathbf{Y}\phi \mid \phi\mathbf{U}\phi \mid \phi\mathbf{S}\phi$$

where  $\alpha_i$ 's are a.t.t.'s,  $R \in \mathcal{R}$ ,  $\mathbf{X}$  and  $\mathbf{Y}$  are the usual “next” and “previous” operators from  $LTL$ , as well as the usual “until”  $\mathbf{U}$  and “since”  $\mathbf{S}$  operators. Notice that  $\mathbf{X}$  and  $\mathbf{X}$  are two distinct operators, with similar meaning. If  $\phi$  is a formula,  $\mathbf{X}\phi$  has the known meaning as in PLTL, while  $\mathbf{X}\alpha$ , where  $\alpha$  is an a.t.t., denotes the *value* of  $\alpha$  in the next time instant. The same holds for  $\mathbf{Y}$  and  $\mathbf{Y}$ , which refer to the previous time instant. Thanks to the obvious property that, for each  $h \geq k$ ,  $\mathbf{X}^h\mathbf{Y}^k x \equiv \mathbf{X}^{h-k}x$  and  $\mathbf{Y}^h\mathbf{X}^k x \equiv \mathbf{Y}^{h-k}x$ , in the following we will assume, with no loss of generality, that a.t.t.'s do not contain any nested alternated occurrences of the operators  $\mathbf{X}$  and  $\mathbf{Y}$ . Each relation symbol is associated with a nonnegative integer denoting its arity. As we will see in Section 5, we can treat separately 0-ary relations, whose set is denoted by  $\mathcal{R}_0$ . We also write  $\text{CLTLB}(\mathcal{D}, \mathcal{R}_0)$  to denote the language CLTLB over the constraint system  $\mathcal{D}$  whose 0-ary relations are exactly those in  $\mathcal{R}_0$ .  $\text{CLTL}(\mathcal{D})$  is the future fragment of  $\text{CLTLB}(\mathcal{D})$  such that only  $\mathbf{X}$ ,  $\mathbf{X}$  and  $\mathbf{U}$  occur in formulae.

The *depth*  $|\alpha|$  of an a.t.t. is the total amount of temporal shift needed in evaluating  $\alpha$ :

$$|x| = 0, \quad |\mathbf{X}\alpha| = |\alpha| + 1, \quad |\mathbf{Y}\alpha| = |\alpha| - 1.$$

Let  $\phi$  be a  $\text{CLTLB}(\mathcal{D}, \mathcal{R}_0)$  formula,  $x$  a variable of  $V$  and  $\Gamma_x(\phi)$  the set of all a.t.t.'s occurring in  $\phi$  in which  $x$  appears. We define the “look-forwards”  $\lceil\phi\rceil_x$  and “look-backwards”  $\lfloor\phi\rfloor_x$  of  $\phi$  relatively to  $x$  as:

$$\lceil\phi\rceil_x = \max_{\alpha_i \in \Gamma_x(\phi)} \{0, |\alpha_i|\}, \quad \lfloor\phi\rfloor_x = \min_{\alpha_i \in \Gamma_x(\phi)} \{0, |\alpha_i|\}.$$

The definitions above naturally extend to  $V$  by letting  $\lceil\phi\rceil = \max_{x \in V} \{\lceil\phi\rceil_x\}$ ,  $\lfloor\phi\rfloor = \min_{x \in V} \{\lfloor\phi\rfloor_x\}$ . Hence,  $\lceil\phi\rceil$  ( $\lfloor\phi\rfloor$ ) is the largest (smallest) depth of all the a.t.t.'s of  $\phi$ , representing the length of the future (past) segment needed to evaluate  $\phi$  in the current instant.

### 2.3. Semantics

The semantics of  $\text{CLTLB}(\mathcal{D}, \mathcal{R}_0)$  formulae is defined with respect to a strict linear order representing time  $(\mathbb{Z}, <)$ . Truth values of propositions in  $\mathcal{R}_0$ , and values of variables belonging to  $V$  are defined by a pair  $(\pi, \sigma)$  where  $\sigma : \mathbb{Z} \times V \rightarrow D$  is a function which defines the value of variables at each position in  $\mathbb{Z}$  and  $\pi : \mathbb{Z} \rightarrow \wp(\mathcal{R}_0)$  is a function associating a subset of the set of propositions with each element of  $\mathbb{Z}$ . The value of terms is defined with respect to  $\sigma$  as follows:

$$\sigma(i, \alpha) = \sigma(i + |\alpha|, x_\alpha)$$

assuming that  $x_\alpha$  is the variable in  $V$  occurring in term  $\alpha$ . The semantics of a  $\text{CLTLB}(\mathcal{D}, \mathcal{R}_0)$  formula  $\phi$  at instant  $i \geq 0$  over a linear structure  $(\pi, \sigma)$  is recursively defined by means of a satisfaction relation  $\models$  as follows, for every formulae  $\phi, \psi$  and for every a.t.t.  $\alpha$ :

$$\begin{aligned} (\pi, \sigma), i &\models p \Leftrightarrow p \in \pi(i) \text{ for } p \in \mathcal{R}_0 \\ (\pi, \sigma), i &\models R(\alpha_1, \dots, \alpha_n) \Leftrightarrow (\sigma(i + |\alpha_1|, x_{\alpha_1}), \dots, \sigma(i + |\alpha_n|, x_{\alpha_n})) \in R \\ (\pi, \sigma), i &\models \neg\phi \Leftrightarrow (\pi, \sigma), i \not\models \phi \\ (\pi, \sigma), i &\models \phi \wedge \psi \Leftrightarrow (\pi, \sigma), i \models \phi \text{ and } (\pi, \sigma), i \models \psi \\ (\pi, \sigma), i &\models \mathbf{X}\phi \Leftrightarrow (\pi, \sigma), i + 1 \models \phi \\ (\pi, \sigma), i &\models \mathbf{Y}\phi \Leftrightarrow (\pi, \sigma), i - 1 \models \phi \wedge i > 0 \\ (\pi, \sigma), i &\models \phi\mathbf{U}\psi \Leftrightarrow \exists j \geq i : (\pi, \sigma), j \models \psi \wedge (\pi, \sigma), n \models \phi \forall i \leq n < j \\ (\pi, \sigma), i &\models \phi\mathbf{S}\psi \Leftrightarrow \exists 0 \leq j \leq i : (\pi, \sigma), j \models \psi \wedge (\pi, \sigma), n \models \phi \forall j < n \leq i \end{aligned}$$

where  $x_{\alpha_i}$  is the variable that appears in  $\alpha_i$ , and  $R \in \mathcal{R} \setminus \mathcal{R}_0$ .

Notice that  $X$  and  $\mathbf{X}$  are two distinct operators, with similar meaning. If  $\phi$  is a formula,  $\mathbf{X}\phi$  has the known meaning as in PLTL, while  $X\alpha$ , where  $\alpha$  is an a.t.t., denotes the *value* of  $\alpha$  in the next time instant. The same holds for  $Y$  and  $\mathbf{Y}$  but they refer to the previous position over time.

A formula  $\phi \in \text{CLTLB}(\mathcal{D}, \mathcal{R}_0)$  is *satisfiable* if there exists a pair  $(\pi, \sigma)$  such that  $(\pi, \sigma), 0 \models \phi$ ; in this case, we say that  $(\pi, \sigma)$  is a *model* of  $\phi$ ,  $\pi$  is a *propositional model* and  $\sigma$  is an *arithmetic model*. By introducing as primitive the connective  $\vee$ , the dual operators “release”  $\mathbf{R}$ , “trigger”  $\mathbf{T}$  and “previous”  $\mathbf{Z}$  are defined as:  $\phi \mathbf{R} \psi \equiv \neg(\neg\phi \mathbf{U} \neg\psi)$ ,  $\phi \mathbf{T} \psi \equiv \neg(\neg\phi \mathbf{S} \neg\psi)$  and  $\mathbf{Z}\phi \equiv \neg\mathbf{Y}\neg\phi$ ; by applying De Morgan’s rules, we may assume every CLTLB formula to be in positive normal form, i.e., negation may only occur in front of atomic propositions while negated  $\mathcal{D}$ -constraints are avoided by means of their complement relations.

## 2.4. CLTL with automata

In this section, we recall some known results where the propositional part  $\pi$  of  $(\pi, \sigma)$  is either missing or can be eliminated. It is proved that, for some constraint system  $\mathcal{D}$  more expressive than  $\text{IPC}^*$ , the future fragment  $\text{CLTL}(\mathcal{D})$  can encode runs of a class of Turing equivalent two-counter automata called Minsky machines. Minsky machines are finite state automata endowed with two positive integer counters  $c_1, c_2$  which can be either incremented or decremented by 1 and tested against 0 over transitions. Any formalism which is able to simulate such class of machines inherits the full expressiveness and undecidability properties of Turing machines. To represent increment and decrement instructions the grammar of formulae  $\xi$  of  $\text{IPC}^*$  must be enriched with formulae of the form  $x < y + d$ , where  $d \in \mathcal{D}$  and  $x, y$  are two variables. Such a language is called *difference logic*, which we denote by  $\text{DL}^+$  (DL is the fragment without modulo operator  $\equiv_b$ ). Hereafter, we write  $\text{CLTL}_a^b(\mathcal{D})$  to denote the language of CLTL formulae such that the cardinality of set  $V$  of variables is  $a$  and the length  $|\phi|$  is equal to  $b$  ( $|\phi| = 0$ ).

The first undecidability result for the satisfiability of CLTL is given by Comon and Cortier [Comon and Cortier 2000, Theorem 3] who show that halting runs of a Minsky machine can be encoded into  $\text{CLTL}_3^1(\text{DL})$  formulae where one auxiliary counter encodes control states of the system labeling instructions. Therefore, the satisfiability problem for  $\text{CLTL}_3^1(\text{DL})$  is  $\Sigma_1^1$ -hard. The authors suggest a way to regain decidability by means of a syntactic restriction on formulae including the  $\mathbf{U}$  temporal operator. The “flat” fragment of  $\text{CLTL}_\omega^1(\text{DL})$  consists of CLTL formulae such that subformula  $\phi$  of  $\phi \mathbf{U} \psi$  is  $\top$ ,  $\perp$  or a conjunction  $\zeta_1 \wedge \dots \wedge \zeta_m$  where  $\zeta_i \in \text{DL}$ . The fragment has a nice correspondence with a special class of counter system (flat relational counter system) with Büchi acceptance condition for which the emptiness problem is decidable. Satisfiability is undecidable also in the case of  $\text{CLTL}_1^2(\text{DL})$  and  $\text{CLTL}_2^2(\text{DL})$ . In fact, even though  $\text{CLTL}_1^2(\text{DL})$  has only one variable, it is expressive enough to encode runs of Minsky machines. Models of  $\text{CLTL}_1^2(\text{DL})$  formulae can represent counter  $c_1$  at even positions and counter  $c_2$  at odd positions. The recurrence problem for nondeterministic Minsky machines, which is  $\Sigma_1^1$ -hard [Alur and Henzinger 1994], can be reduced to the satisfiability problem for  $\text{CLTL}_1^2(\text{DL})$ , which then results to be  $\Sigma_1^1$ -hard. From previous undecidability results, the satisfiability problem for the CLTL language over two integer variables  $\text{CLTL}_2^1(\text{DL})$  is  $\Sigma_1^1$ -hard. Formulae of  $\text{CLTL}_1^2(\text{DL})$  can be syntactically translated to formulae of  $\text{CLTL}_2^1(\text{DL})$  by means of a map  $f$  such that  $\phi$  belonging to  $\text{CLTL}_1^2(\text{DL})$  is satisfiable if, and only if,  $f(\phi)$  belonging to  $\text{CLTL}_2^1(\text{DL})$  is satisfiable. Both the languages  $\text{CLTL}_1^2(\text{DL})$  and  $\text{CLTL}_2^1(\text{DL})$  are also  $\Sigma_1^1$ -complete by reducing the  $\Sigma_1^1$ -hard model-checking problem to satisfiability.

The satisfiability (and model-checking) problem for CLTL over structure  $(D, <, =)$  with  $D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$  is studied in [Demri and D’Souza 2007], and for  $\text{IPC}^*$  in [Demri and Gascon 2007]. Decidability of the satisfiability problem for the above cases is shown by means of an automata-based approach similar to the standard case for LTL. Satisfiability for  $\text{CLTL}_\omega^\omega(\text{IPC}^*)$  and  $\text{CLTL}_\omega^\omega(<, =)$  over  $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$  is obtained by Demri and Gascon in [Demri and Gascon 2005] by reducing the problem to the emptiness problem for Büchi automata. Given a CLTL formula  $\phi$ , it is possible to define an automaton  $\mathcal{A}_\phi$  such that  $\phi$  is satisfiable if, and only if,  $\mathcal{L}(\mathcal{A}_\phi)$  is not empty.

Since the emptiness of  $\mathcal{L}(\mathcal{A}_\phi)$  in the considered structures is decidable with PSPACE upper bound (in the dimension of  $\phi$ ), then the satisfiability problem is also decidable with the same complexity.

Hereafter, we restrict  $\mathcal{D}$  to be the structure defined by  $\text{IPC}^*$ , or by  $(D, <, =)$ , where  $D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$ . We briefly recall some useful notions which we use in the following sections and which are essential to develop our decision procedure without automata construction. We will simply write CLTL to denote  $\text{CLTL}_\omega^\omega$ . In order to represent exactly models of a  $\text{CLTLB}(\mathcal{D})$  formula  $\phi$  (without the Y temporal modality over variables) by means of automata, we need to represent symbolically all models  $\sigma$  such that  $\sigma \models \phi$ .

Let  $\phi$  be a  $\text{CLTLB}(\mathcal{D})$  formula and  $\text{terms}(\phi)$  be the set of arithmetic terms of the form  $X^i x$  for all  $0 \leq i \leq \lceil \phi \rceil$  and for all  $x \in V$ . If domain  $D$  is discrete, let  $\text{const}(\phi) = \{m, \dots, M\}$  be the set of constants occurring in  $\phi$ , where  $m, M \in D$  are the minimum and maximum constants. We extend  $\text{const}(\phi)$  to the set  $\text{const}'(\phi) = [m, M]$  of all values between  $m$  and  $M$ . Constant  $K$  is the l.c.m. of constants occurring in periodic constraints  $x \equiv_c y$  and  $x \equiv_c y + d$ .

A set of  $\mathcal{D}$ -constraints over  $\text{terms}(\phi)$  is *maximally consistent* if for every  $\mathcal{D}$ -constraint  $\theta$  over  $\text{terms}(\phi) \cup \text{const}(\phi)$ , either  $\theta$  or  $\neg\theta$  is in the set.

**Definition 2.1.** A *symbolic valuation*  $sv$  for  $\phi$  is a maximally consistent set of  $\mathcal{D}$ -constraints over  $\text{terms}(\phi)$  and  $\text{const}(\phi)$ .

The original definition of symbolic valuation for  $\text{IPC}^*$  constraint systems in [Demri and Gascon 2005] is slightly different. There, it is defined as triple  $\langle S_1, S_2, S_3 \rangle$  where  $S_1$  is a maximally consistent set of  $\mathcal{D}$ -constraints over  $\text{terms}(\phi)$  and  $\text{const}(\phi)$ ;  $S_2$  is a set of constraints of the form  $x = d$  and  $S_3$  is a set of periodic constraints  $x \equiv_K c$ . Our definition of symbolic valuations does not consider sets  $S_2$  and  $S_3$  because they are inherently represented by the  $k$ -bounded arithmetical models  $\sigma_k$  defined in Section 3. In other words, “explicit” assignments to variables from  $\sigma_k$  do not require to be symbolically represented by a symbolic constraint of the form  $x = d$ .

The set of all symbolic valuations for  $\phi$  is denoted by  $SV(\phi)$ . To define the satisfiability of a symbolic valuation, each a.t.t. is considered as a new fresh variable. Let  $A$  be a set of variables and  $f : \text{terms}(\phi) \rightarrow A$  an injective function mapping each a.t.t. of  $\phi$  into a fresh variable in set  $A$ . Function  $f$  is naturally extended to every symbolic valuation  $sv$  for  $\phi$ , by replacing each a.t.t.  $\alpha \in \text{terms}(\phi)$  in  $sv$  with  $f(\alpha)$ . Symbolic valuations for  $\phi$  are now defined over the set  $f(\text{terms}(\phi))$ . A symbolic valuation  $sv$  for  $\phi$  is *satisfiable* if there exists a  $\mathcal{D}$ -valuation  $v' : A \rightarrow D$ , such that  $v' \models_{\mathcal{D}} f(sv)$ , i.e., satisfiability of  $sv$  considers all a.t.t.’s as fresh variables.

Given a symbolic valuation  $sv$  and a  $\mathcal{D}$ -constraint  $\xi$  over a.t.t.’s, we write  $sv \models^{\text{sym}} \xi$  if for every  $\mathcal{D}$ -valuation  $v'$  such that  $v' \models_{\mathcal{D}} f(sv)$  then  $v' \models_{\mathcal{D}} \xi$ . We assume that the problem of checking  $sv \models^{\text{sym}} \xi$  is decidable. The satisfaction relation  $\models^{\text{sym}}$  can also be extended to sequences of symbolic valuations; it is the same as  $\models$  for all temporal operators except for atomic formulae:

$$\rho, i \models^{\text{sym}} \xi \Leftrightarrow \rho(i) \models^{\text{sym}} \xi.$$

Then, given a  $\text{CLTLB}(\mathcal{D})$  formula  $\phi$ , we say that  $\rho$  *symbolically satisfies*  $\phi$  (or  $\rho$  is a *symbolic model* for  $\phi$ ) when  $\rho, 0 \models^{\text{sym}} \phi$ .

**Definition 2.2.** A pair of symbolic valuations  $(sv_1, sv_2)$  for  $\phi$  is *locally consistent* if, for all  $R$  in  $\mathcal{D}$ :

$$R(X^{i_1}x_1, \dots, X^{i_n}x_n) \in sv_1 \text{ implies } R(X^{i_1-1}x_1, \dots, X^{i_n-1}x_n) \in sv_2$$

with  $i_j \geq 1$  for all  $j \in [1, n]$ . A sequence of symbolic valuations  $sv_0, sv_1, \dots$  is *locally consistent* if all pairs  $(sv_i, sv_{i+1})$ ,  $i \geq 0$ , are locally consistent.

A locally consistent infinite sequence  $\rho : \mathbb{N} \rightarrow SV(\phi)$  of symbolic valuations *admits an arithmetic model*, if there exists a  $\mathcal{D}$ -valuation sequence  $\eta$  such that  $\eta, i \models \rho(i)$ , for all  $i \geq 0$ . In this case, we write  $\eta, 0 \models \rho$ .

The following fundamental proposition draws a link between the satisfiability by sequences of symbolic valuations and by sequences of  $\mathcal{D}$ -valuations.

**PROPOSITION 2.3** ([DEMRI AND D’SOUZA 2007]). *A CLTL( $\mathcal{D}$ ) formula  $\phi$  is satisfiable if, and only if, there exists a symbolic model for  $\phi$  which admits an arithmetical model, i.e., there exists  $\rho$  and  $\sigma$  such that  $\rho, 0 \models^{\text{sym}} \phi$  and  $\sigma, 0 \models \rho$ .*

Following [Demri and D’Souza 2007], for constraint systems of the form  $(D, <, =)$ , where  $<$  is a strict total ordering on  $D$ , it is possible to represent a symbolic valuation  $sv$  by its labeled directed graph  $G_{sv} = \{V, \tau \subseteq V \times \{<, =\} \times V\}$ , such that  $(x, \sim, y) \in \tau$  if, and only if,  $x \sim y \in sv$ . This construction extends also to sequences: given a sequence  $\rho$  of symbolic valuations, it is possible to represent  $\rho$  via the graph  $G_\rho$  obtained by superimposition of the graphs corresponding to the symbolic evaluations  $\rho(i)$ . More formally  $G_\rho = (V \times \mathbb{N}, \tau_\rho)$ , where  $((x, i), \sim, (y, j)) \in \tau_\rho$  if, and only if, either  $i \leq j$  and  $(x \sim X^{j-i}y) \in \rho(i)$ , or  $i > j$  and  $(X^{i-j}x \sim y) \in \rho(j)$ .

An infinite path  $d : \mathbb{N} \rightarrow V \times \mathbb{N}$  over  $G_\rho$ , is called a *forward* (resp. *backward*) path if:

- (1) for all  $i \in \mathbb{N}$ , there is an edge from  $d(i)$  to  $d(i+1)$  (resp., an edge from  $d(i+1)$  to  $d(i)$ );
- (2) for all  $i \in \mathbb{N}$ , if  $d(i) = (x, j)$  and  $d(i+1) = (x', j')$ , then  $j \leq j'$ .

A forward (resp. backward) path is *strict* if there exist infinitely many  $i$  for which there is a  $<$ -labeled edge from  $d(i)$  to  $d(i+1)$  (resp., from  $d(i+1)$  to  $d(i)$ ). Intuitively, a (strict) forward path represents a sequence of (strict) monotonic increasing values whereas a (strict) backward path represents a sequence of (strict) monotonic decreasing values.

Given a CLTL( $\mathcal{D}$ ) formula  $\phi$ , it is possible [Demri and D’Souza 2007] to define a Büchi automaton  $\mathcal{A}_\phi$  recognizing symbolic models of  $\phi$ , and then reducing the satisfiability of  $\phi$  to the emptiness of  $\mathcal{A}_\phi$ . The idea is that automaton  $\mathcal{A}_\phi$  should accept the intersection of the following languages, which defines exactly the language of symbolic models of  $\phi$ :

- (1) the language of LTL models  $\rho$ ;
- (2) the language of sequences of locally consistent symbolic valuations;
- (3) the language of sequences of symbolic valuations which admit an arithmetic model.

Language (1) is accepted by the Vardi-Wolper automaton  $\mathcal{A}_s$  of  $\phi$  ([Vardi and Wolper 1986]), while language (2) is recognized by the automaton  $\mathcal{A}_\ell = (SV(\phi), sv_0, \rightarrow, SV(\phi))$ , where  $sv_i \xrightarrow{sv_i} sv_{i+1}$  if, and only if, all pairs  $(sv_i, sv_{i+1})$  are locally consistent ([Demri and D’Souza 2007]).

If the constraint system we are considering has the *completion property* (defined next), then any sequences of locally consistent symbolic valuations admit an arithmetic model, and condition (3) reduces to (2).

**2.4.1. Completion property.** Each automaton involved in the definition of  $\mathcal{A}_\phi$  has the function of “filtering” sequences of symbolic valuations so that 1) they are locally consistent, 2) they satisfy an LTL property and 3) they admit a (arithmetic) model. For some constraint systems, admitting a model is a consequence of local consistency. A set of relations over  $D$  has the *completion property* if, given:

- (i). a symbolic valuation  $sv$  over a finite set of variables  $H \subseteq V$ ,
- (ii). a subset  $H' \subseteq H$ ,
- (iii). a valuation  $v'$  over  $H'$  such that  $v' \models sv'$ , where  $sv'$  is the subset of atomic formulae in  $sv$  which uses only variables in  $H'$

then there exists a valuation  $v$  over  $V$  extending  $v'$  such that  $v \models sv$ . An example of such a relational structure is  $(\mathbb{R}, <, =)$ . Let  $(D, <, =)$  be a relational structure defining the language of atomic formulae. We say that  $D$  is *dense*, with respect to the order  $<$ , if for each  $d, d' \in D$  such that  $d < d'$ , there exists  $d'' \in D$  such that  $d < d'' < d'$ . Whereas  $D$  is said to be *open* when for each  $d \in D$ , there exist two elements  $d', d'' \in D$  such that  $d' < d < d''$ .

**LEMMA 2.4** (LEMMA 5.3, [DEMRI AND D’SOUZA 2007]). *Let  $(D, <, =)$  be a relational structure where  $D$  is infinite and  $<$  is a total order. Then, it satisfies the completion property if, and only if, domain  $D$  is dense and open.*

The following result relies on the fact that every locally consistent sequence of symbolic valuations with respect to the relational structure  $\mathcal{D}$  admits a model.

**PROPOSITION 2.5.** *Let  $\mathcal{D}$  be a relational structure satisfying the completion property and  $\phi$  be a  $\text{CLTL}(\mathcal{D})$  formula. Then, the language of sequences of symbolic valuations which admit a model is  $\omega$ -regular.*

In this case the automaton  $\mathcal{A}_\phi$  that recognizes exactly all the sequences of symbolic valuations which are symbolic models of  $\phi$  is defined by the intersection (*à la* Büchi)  $\mathcal{A}_\phi = \mathcal{A}_s \cap \mathcal{A}_\ell$ .

In general, however, language (3) may *not* be  $\omega$ -regular. Nevertheless, if the constraint system is of the form  $(D, <, =)$ , it is possible to define an automaton  $\mathcal{A}_C$  that accepts a superset of language (3), but such that all its *ultimately periodic* words are sequences of symbolic valuations that admit an arithmetic model. Actually,  $\mathcal{A}_C$  recognizes a sequence  $\rho$  of symbolic valuations that satisfies the following property:

**PROPERTY 2.6 (C).** *There do not exist vertices  $u$  and  $v$  in the same symbolic valuation in  $G_\rho$  satisfying all the following conditions:*

- (1) *there is an infinite forward path  $d$  from  $u$ ;*
- (2) *there is an infinite backward path  $e$  from  $v$ ;*
- (3)  *$d$  or  $e$  is strict;*
- (4) *for each  $i, j \in \mathbb{N}$ , whenever  $d(i)$  and  $e(j)$  belong to the same symbolic valuation there exists an edge, labeled by  $<$ , from  $d(i)$  to  $e(j)$ .*

Informally, property *C* guarantees that in the model there does not exist an infinite forward path whose values are infinitely often less than values of an infinite backward path; in other words, an infinite strict/non-strict monotonic increasing sequence of values can not be infinitely often less than an infinite non-strict/strict monotonic decreasing sequence of values.

The proposed method is general and it can be used whenever it is possible to build an automaton  $\mathcal{A}_C$  which defines a condition *C* guaranteeing the existence of a sequence  $\sigma$  such that  $\sigma, 0 \models \rho$ . In particular, for constraint systems  $\text{IPC}^*$ ,  $(\mathbb{N}, <, =)$ , and  $(\mathbb{Z}, <, =)$ ,  $\mathcal{A}_C$  can effectively be built. If  $\mathcal{A}_\phi$  is defined as the (Büchi) product of  $\mathcal{A}_\ell$ ,  $\mathcal{A}_s$ ,  $\mathcal{A}_C$ , and, since emptiness of Büchi automata can be checked just on ultimately periodic words, the language of  $\mathcal{A}_\phi$  is empty if, and only if,  $\phi$  has not a symbolic model.

When the condition *C* is sufficient and necessary for the existence of models  $\sigma$  such that  $\sigma, 0 \models \rho$ , then automaton  $\mathcal{A}_\phi$  represents all sequences of symbolic valuations which admit a model  $\sigma$ . A fundamental lemma, on which Proposition 2.8 below relies, draws a sufficient and necessary condition for the existence of models of sequences of symbolic valuations.

**LEMMA 2.7** ([DEMRI AND D'SOUZA 2007]). *Let  $\rho$  be an  $\omega$ -periodic sequence of symbolic valuations of the form  $\rho = \alpha\beta^\omega$  that is locally consistent. Then  $\rho$  admits a model  $\sigma$  if, and only if,  $\rho$  satisfies *C*.*

Therefore, the satisfiability problem can be solved by checking the emptiness of the language recognized by the automaton  $\mathcal{A}_\phi$ .

**PROPOSITION 2.8** ([DEMRI AND D'SOUZA 2007]). *A  $\text{CLTL}(\mathcal{D})$  formula is satisfiable iff the language  $\mathcal{L}(\mathcal{A}_\phi)$  is not empty.*

The next section provides two syntactic translations needed to obtain, from  $\text{CLTLB}(\mathcal{D}, \mathcal{R}_0)$  formulae, equisatisfiable  $\text{CLTLB}(\mathcal{D})$  formulae without occurrences of the temporal modality *Y* and of 0-ary relations. The two reductions are essential to take advantage of Proposition 2.5 and Lemma 2.7 which allow us to define the decision procedure based on “bounded” satisfiability of Section 3. In particular, we define the bounded satisfiability problem which consists in looking for ultimately periodic symbolic models  $\alpha\beta^\omega$  such that prefix  $\alpha\beta$  is of fixed length (which is provided as input of problem); moreover, we require that  $\alpha\beta$  admits a *finite* model  $\sigma_k$ . Therefore, we show that when a



formula  $\phi$  is boundedly satisfiable then it is also satisfiable. We provide a (linear-space) reduction from the bounded satisfiability problem to the satisfiability of formulae in the quantifier-free theory of equality and uninterpreted functions QF-EUF combined with  $\mathcal{D}$ .

### 3. BOUNDED SATISFIABILITY PROBLEM

In this section, we provide the definition of the  $k$ -satisfiability problem for CLTLB( $\mathcal{D}$ ) formulae in terms of the existence of a so-called  $k$ -bounded arithmetical model  $\sigma_k$ , which provides a finite representation of infinite symbolic models by means of ultimately periodic words. This allows us to prove that  $k$ -satisfiability is still representative of the satisfiability problem as defined in Section 2.3. In fact, for some constraint systems, a bounded solution can be used to build the infinite model  $\sigma$  for the formula from the  $k$ -bounded one  $\sigma_k$  and from its symbolic model. We show that a formula  $\phi$  is satisfiable if, and only if, it is  $k$ -satisfiable and its bounded solution  $\sigma_k$  can be used to derive its infinite model  $\sigma$ . In case of negative answer to a  $k$ -bounded instance, we can not immediately entail the unsatisfiability of the formula. However, we prove that for every formula  $\phi$  there exists an upper bound  $K$ , which can effectively be determined, such that if  $\phi$  is not  $k$ -satisfiable for all  $k$  in  $[1, K]$  then  $\phi$  is unsatisfiable.

We first define the Bounded Satisfiability Problem (BSP), by considering bounded symbolic models of CLTLB( $\mathcal{D}$ ) formulae. A bounded symbolic model is, informally, a finite representation of infinite CLTLB( $\mathcal{D}$ ) models over the alphabet of symbolic valuations  $SV(\phi)$ . We restrict the analysis to ultimately periodic symbolic models, i.e., of the form  $\rho = \alpha(\beta)^\omega$ . BSP is defined with respect to a  $k$ -bounded model  $\sigma_k : \{ \lfloor \phi \rfloor, \dots, k + \lceil \phi \rceil \} \times V \rightarrow D$ , a finite sequence  $\rho'$  (with  $|\rho'| = k + 1$ ) of symbolic valuations and a  $k$ -bounded satisfaction relation  $\models_k$  defined as follows:

$$\sigma_k, 0 \models_k \rho' \text{ iff } \sigma_k, i \models \rho'(i) \text{ for all } 0 \leq i \leq k.$$

The  $k$ -satisfiability problem of formula  $\phi$  is defined as follows:

*Input.* A CLTLB( $\mathcal{D}$ ) formula  $\phi$ , a constant  $k \in \mathbb{N}$

*Problem.* Is there an ultimately periodic sequence of symbolic valuations  $\rho = \alpha(\beta)^\omega$  (with  $|\alpha\beta| = k + 1$ ), such that  $\rho, 0 \models^{sym} \phi$  and which admits a  $k$ -bounded model  $\sigma_k$  such that  $\sigma_k \models_k \rho'$ , with  $\rho' = \alpha\beta$ ?

Since the length  $k$  is fixed, the procedure for determining the satisfiability of CLTLB( $\mathcal{D}$ ) formulae over bounded models is not complete: even if there is no accepting run of automaton  $\mathcal{A}_\phi$  when  $\rho'$  as above has length  $k$ , there may be accepting runs for a larger  $\rho'$ .

*Definition 3.1.* Given a CLTLB( $\mathcal{D}$ ) formula  $\phi$ , its *completeness threshold*  $K_\phi$ , if it exists, is the smallest number such that  $\phi$  is satisfiable if and only if  $\phi$  is  $K_\phi$ -satisfiable.

### 4. AN ENCODING FOR BSP WITHOUT AUTOMATA

In this section, we prove that the BSP for a CLTLB( $\mathcal{D}$ ) formula can be reduced to the satisfiability of a quantifier-free formula in the theory  $\text{EUF} \cup \mathcal{D}$  (QF-EUD), where EUF is the theory of Equality and Uninterpreted Functions, provided that  $\mathcal{D}$  includes a copy of  $\mathbb{N}$  with the successor relation and that  $\text{EUF} \cup \mathcal{D}$  is consistent. The last condition is easily verified in the case of the union of two consistent, disjoint, stably infinite theories (as is the case for EUF and arithmetic). In [Bersani et al. 2010] a similar approach is described for the case of Integer Difference Logic (DL) constraints. It is worth noting that standard LTL can be encoded by a formula in QF-EUD with  $\mathcal{D} = (\mathbb{N}, <)$ . In this case, the encoding is more succinct than the Boolean one proposed in [Biere et al. 2006]. The encoding presented below represents ultimately periodic sequences of symbolic valuations  $\rho$  of the form  $sv_0sv_1 \dots sv_{loop-1}(sv_{loop} \dots sv_k)^\omega$ . To do this, we use a positive integer variable *loop* for which we require  $sv_{loop-1} = sv_k$ . Therefore, we look for a finite word  $\rho' = sv_0sv_1 \dots sv_{loop-1}(sv_{loop} \dots sv_k)sv_{loop}$  of length  $k + 2$  representing the ultimately periodic model above. Instant  $k + 1$  in the encoding is used to correctly represent the periodicity of  $\rho$  by

constraining atomic formulae (propositions and relations) at positions  $loop$  and  $k + 1$ . Moreover, all subformulae of  $\phi$  at positions  $loop - 1$  and  $k$  must be the same.

*Encoding terms.* We introduce an *arithmetic formula function* to encode all terms in the set  $terms(\phi)$ . To do so an uninterpreted function  $\alpha : \mathbb{Z} \rightarrow D$  is associated with each arithmetic temporal term  $\alpha \in terms(\phi)$ . Let  $\alpha$  be such a term, then the arithmetic formula function associated with it (denoted by the same name but written in boldface), is recursively defined with respect to a finite sequence of valuations  $\sigma_k$  as:

$$\frac{\alpha \quad \begin{array}{c|c} 0 \leq i < k & i = k \\ \hline x & \mathbf{x}(i) = \sigma_k(i, x) \\ X\alpha' & \mathbf{\alpha}(i) = \alpha'(i + 1) \end{array} \quad \begin{array}{c|c} i = k & \\ \hline \mathbf{x}(k) = \sigma_k(k, x) \\ \mathbf{\alpha}(k) = \sigma_k(k + |\alpha'| + 1, x_{\alpha'}) \end{array}}{\frac{\alpha \quad \begin{array}{c|c} 0 < i \leq k + 1 & i = 0 \\ \hline Y\alpha' & \mathbf{\alpha}(i) = \alpha'(i - 1) \end{array} \quad \begin{array}{c|c} i = 0 & \\ \hline \mathbf{\alpha}(0) = \sigma_k(|\alpha'| - 1, x_{\alpha'}) \end{array}}$$

Conjunction of the above subformulae gives the formula  $|ArithConstraints|_k$ . Implementing  $|ArithConstraints|_k$  is straightforward. In fact, the assignments of values to variables are defined by the interpretation of the symbols of the QF-EUD formula. The values of variables  $x$  at positions before 0 and  $k$ , i.e. in intervals  $[\lfloor \phi \rfloor, -1]$  and  $[k + 1, k + \lceil \phi \rceil]$ , are defined by means of the values of terms  $\alpha = X^i x$  and  $\alpha = Y^i x$ . For instance, the value of  $x$  at position  $0 > i \geq \lfloor \phi \rfloor$  is  $\sigma_k(i, x)$  but it is defined by the assignment for term  $\alpha = Y^i x$  at position 0.

*Encoding relations.* The formula  $|PropConstraints|_k$  encodes atomic subformulae  $\theta$  containing relations over a.t.t.'s. Let  $R$  be an  $n$ -ary relation of  $\mathcal{R}$  that appears in  $\phi$ , and  $\alpha_1, \dots, \alpha_n$  be a.t.t.'s. We introduce a formula predicate  $\theta : \mathbb{N} \rightarrow \{true, false\}$  for all  $R(\alpha_1, \dots, \alpha_n)$  in  $\phi$ :

$$\frac{\theta \quad \begin{array}{c|c} 0 \leq i \leq k + 1 \\ \hline R(\alpha_1, \dots, \alpha_n) \end{array}}{\theta(i) \Leftrightarrow R(\mathbf{\alpha}_1(i), \dots, \mathbf{\alpha}_n(i))}$$

*Encoding formulae.* The truth value of a CLTLB formula is defined with respect to the truth value of its subformulae. We associate with each subformula  $\theta$  a *formula predicate* that is a unary uninterpreted predicate (denoted by the same name but written in boldface)  $\theta : \mathbb{N} \rightarrow \{true, false\}$ . When the subformula  $\theta$  holds at instant  $i$  then  $\theta(i)$  holds. As the length of paths is fixed to  $k + 1$  and all paths start from 0, formula predicates are actually subsets of  $\{0, \dots, k + 1\}$ . Let  $\theta$  be a subformula of  $\phi$ , formula predicate  $\theta$  is recursively defined as:

$$\frac{\theta \quad \begin{array}{c|c} 0 \leq i \leq k + 1 \\ \hline \neg\psi & \mathbf{\theta}(i) \Leftrightarrow \neg\mathbf{\psi}(i) \\ \psi_1 \wedge \psi_2 & \mathbf{\theta}(i) \Leftrightarrow \mathbf{\psi}_1(i) \wedge \mathbf{\psi}_2(i) \end{array}}{\mathbf{\theta}(i) \Leftrightarrow \mathbf{\psi}_1(i) \wedge \mathbf{\psi}_2(i)}$$

Then, the conjunction of the formulae above is also part of formula  $|PropConstraints|_k$ . The temporal behavior of future and past operators is defined by using their traditional fixpoint characterizations:

$$\frac{\theta \quad \begin{array}{c|c} 0 \leq i \leq k \\ \hline X\psi & \mathbf{\theta}(i) \Leftrightarrow \mathbf{\psi}(i + 1) \\ \psi_1 U \psi_2 & \mathbf{\theta}(i) \Leftrightarrow (\mathbf{\psi}_2(i) \vee (\mathbf{\psi}_1(i) \wedge \mathbf{\theta}(i + 1))) \\ \psi_1 R \psi_2 & \mathbf{\theta}(i) \Leftrightarrow (\mathbf{\psi}_2(i) \wedge (\mathbf{\psi}_1(i) \vee \mathbf{\theta}(i + 1))) \end{array}}{\frac{\theta \quad \begin{array}{c|c} 0 < i \leq k + 1 & i = 0 \\ \hline Y\psi & \mathbf{Y}\mathbf{\psi}(i) \Leftrightarrow \mathbf{\psi}(i - 1) \\ \psi_1 S \psi_2 & (\mathbf{\psi}_1 S \mathbf{\psi}_2)(i) \Leftrightarrow (\mathbf{\psi}_2(i) \vee (\mathbf{\psi}_1(i) \wedge (\mathbf{\psi}_1 S \mathbf{\psi}_2)(i - 1))) \\ \psi_1 T \psi_2 & (\mathbf{\psi}_1 T \mathbf{\psi}_2)(i) \Leftrightarrow (\mathbf{\psi}_2(i) \wedge (\mathbf{\psi}_1(i) \vee (\mathbf{\psi}_1 T \mathbf{\psi}_2)(i - 1))) \end{array} \quad \begin{array}{c|c} i = 0 & \\ \hline \perp & (\mathbf{\psi}_1 S \mathbf{\psi}_2)(0) \Leftrightarrow \mathbf{\psi}_2(0) \\ & (\mathbf{\psi}_1 T \mathbf{\psi}_2)(0) \Leftrightarrow \mathbf{\psi}_2(0) \end{array}}$$

The conjunction of the above formulae gives formula  $|TempConstraints|_k$ .

*Encoding periodicity.* To represent ultimately periodic sequences of symbolic valuations we use a positive integer variable  $loop$  encoding periodicity of  $sv_0sv_1 \dots sv_{loop-1}(sv_{loop} \dots sv_k)^\omega$  for which we require  $sv_{loop-1} = sv_k$ . Formula  $|LoopConstraints|_k$  below is defined by means of a *Loop-selecting variable*  $loop \in \mathbb{N}$  that takes values in  $[1, k]$  when the loop exists, which corresponds to the position where the periodic part of  $(sv_{loop} \dots sv_k)$  starts. Let  $\theta$  be an  $n$ -ary relation  $R \in \mathcal{R}$ . Then, periodicity is encoded by the following formula:

$$\bigwedge_{i=1}^k \left( (loop = i) \Rightarrow \bigwedge_{\substack{\theta \in \mathcal{R} \\ \alpha_1, \dots, \alpha_n \in terms(\phi)}} \theta(i-1) = \theta(k) \right).$$

Informally, if the value  $i$  of variable  $loop$  is between 1 and  $k$ , then there exists a loop which starts at  $i$ . The formula  $loop = i$  is well defined in QF-EU( $\mathbb{N}, <$ ).

*Last state constraints* ( $|LastStateConstraints|_k$ ) define an equivalence between truth values at point  $k+1$  and truth values at the point indicated by the  $loop$  variable, since the instant  $k+1$  is representative of the instant  $loop$  along periodic paths. Otherwise, for non-periodic paths, truth values in  $k+1$  are trivially false. These constraints have a similar structure as those in the original Boolean encoding, but here they are defined by only *one* constraint for each subformula  $\theta$  of  $\phi$ , w.r.t. the variable  $loop$ :

$$\left( \bigwedge_{i=1}^k (loop = i) \Rightarrow (\theta(k+1) \Leftrightarrow \theta(i)) \right) \wedge \left( \bigwedge_{i=1}^k \neg(loop = i) \Rightarrow \neg\theta(k+1) \right).$$

*Eventualities for U and R.* To correctly define the semantics of **U** and **R**, their *eventualities* have to be accounted for. Briefly, if  $\psi_1 \mathbf{U} \psi_2$  holds at  $i$ , then  $\psi_2$  eventually holds in some  $j \geq i$ ; if  $\psi_1 \mathbf{R} \psi_2$  does not hold at  $i$ , then  $\psi_2$  eventually does not hold in some  $j \geq i$ . Along finite paths of length  $k$ , eventualities must hold between 0 and  $k$ . Otherwise, if there is a loop, an eventuality may hold within the loop. The original Boolean encoding introduces  $k$  propositional variables for each subformula  $\theta$  of  $\phi$  of the form  $\psi_1 \mathbf{U} \psi_2$  or  $\psi_1 \mathbf{R} \psi_2$  (one for each  $1 \leq i \leq k$ ), which represent the eventuality of  $\psi_2$  implicit in the formula, as first defined in [Biere et al. 2006]. Instead, in the QF-EUD encoding, only *one* variable  $j_{\psi_2} \in D$  is introduced for each  $\psi_2$  occurring in a subformula  $\psi_1 \mathbf{U} \psi_2$  or  $\psi_1 \mathbf{R} \psi_2$ ; let  $\ell$  be a shorthand for  $\bigvee_{i=1}^k (loop = i)$ :

$\theta$	
$\psi_1 \mathbf{U} \psi_2$	$\ell \Rightarrow (\theta(k) \Rightarrow loop \leq j_{\psi_2} \leq k \wedge \psi_2(j_{\psi_2}))$
$\psi_1 \mathbf{R} \psi_2$	$\ell \Rightarrow (\neg\theta(k) \Rightarrow loop \leq j_{\psi_2} \leq k \wedge \neg\psi_2(j_{\psi_2}))$

The conjunction of all the constraints for all the subformulae  $\theta$  of  $\phi$  constitutes the formula  $|Eventually|_k$ .

The complete encoding  $|\phi|_k$  of  $\phi$  consists of the logical conjunction of all above components, together with  $\phi$  evaluated at the first instant of time.

## 5. CORRECTNESS OF THE BSP ENCODING

In this section, we provide a proof of correctness of the encoding defined in Section 4. We split the proof into two parts: with Theorem 5.8 we show that the encoding  $|\phi|_k$  of a formula  $\phi$  represents ultimately periodic runs of automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$  introduced in Section 2.4. In Theorem 5.9 we focus on the fact that  $k$ -satisfiability is strictly related to the existence of ultimately periodic runs of automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$ . Finally, we are able to relate the satisfiability of  $|\phi|_k$  to  $k$ -satisfiability. The next lemmata are useful to prove Theorem 5.8. They are essential for our approach because they state how  $k$ -bounded models  $\sigma_k$  are representative of ultimately periodic sequences of symbolic valuations, which are symbolic models of the formula. In other words, by Lemma 5.3, we can build a

sequence of symbolic valuations from the sequence  $\sigma_k$ . Moreover, if the symbolic sequence satisfies additional constraints enforcing periodicity of relations in  $\mathcal{R}$ , we can derive from  $\sigma_k$  an ultimately periodic symbolic model.

It is worth noting that BSP is defined with respect to sequences of symbolic valuations whereas the encoding considers only atomic subformulae  $R$  occurring in  $\phi$ . By definition, symbolic valuations are maximally consistent sets of  $\mathcal{D}$ -constraints over the set of terms  $terms(\phi)$  occurring in the formula. Maximal consistency of sets guarantees that set  $SV(\phi)$  is a partition of  $D^{|V|}$ . Therefore, any finite sequence of  $\mathcal{D}$ -valuations  $\sigma_k$  induces a finite sequence of symbolic valuations of length  $k$ .

We consider the following assumption  $A_{mc}$ , which guarantees that maximally consistent sets of relations partition the space  $D^n$ , where  $n$  is the cardinality of the set  $V$  of variables.

*Assumption ( $A_{mc}$ ).* For all  $m > 0$  such that there is an  $m$ -ary relation  $R^m \in \mathcal{R}$  and for all  $v \in D^m$ , there exists a unique relation  $R$  such that  $v \models_{\mathcal{D}} R$ .

**LEMMA 5.1.** *Let  $\mathcal{D}$  be a constraint system satisfying assumption ( $A_{mc}$ ),  $\phi$  be a  $CLTLB(\mathcal{D})$  formula and  $v$  be a  $\mathcal{D}$ -valuation extended to terms appearing in symbolic valuations of  $SV(\phi)$ . Then, there is a unique symbolic valuation  $sv$  such that  $v \models_{\mathcal{D}} sv$ .*

**PROOF.** We build a symbolic valuation from the values in  $v$ . We include  $R(\alpha_1, \dots, \alpha_n)$  in  $sv$  when  $v \models_{\mathcal{D}} f(R(\alpha_1, \dots, \alpha_n))$ . We have to show that  $sv$  built in this way is maximally consistent. Consistency is immediate by the fact that if  $v \models_{\mathcal{D}} f(R(\alpha_1, \dots, \alpha_n))$  then it can not hold that  $v \models_{\mathcal{D}} f(\neg R(\alpha_1, \dots, \alpha_n))$ . We prove maximality by showing that if  $sv$  is not maximal then there exists a relation which should not belong to  $sv$  but, by construction, must also be in  $sv$ , thus producing a contradiction. Let us suppose that there is a relation  $R'$  which is not in  $sv$  such that  $v \models_{\mathcal{D}} f(R')$ . Moreover, let us suppose that  $sv \cup \{R'\}$  is consistent; i.e.,  $v \models_{\mathcal{D}} sv \cup \{R'\}$ . By construction, as we include  $R(\alpha_1, \dots, \alpha_n)$  in  $sv$  when  $v \models_{\mathcal{D}} f(R(\alpha_1, \dots, \alpha_n))$ , there is a relation  $R''$ , in  $sv$ , over the same set of terms of  $R'$ . By assumption ( $A_{mc}$ ), we have  $R' = R''$ , because, otherwise, in constraint system  $\mathcal{D}$  we have two different relations,  $R'$  and  $R''$ , over the same set of terms and such that  $v \models_{\mathcal{D}} f(R')$  and  $v \models_{\mathcal{D}} f(R'')$ , where  $v$  is an assignment of values in  $D$  to terms in  $terms(\phi)$ . This contradicts the uniqueness assumption in ( $A_{mc}$ ).  $\square$

**COROLLARY 5.2.** *Let  $\phi$  be a  $CLTLB(\mathcal{D})$  formula,  $v$  a  $\mathcal{D}$ -valuation extended to terms of symbolic valuations and  $sv$  a symbolic valuation in  $SV(\phi)$ . Then, for  $v \models_{\mathcal{D}} sv$  and for all relations  $R \in \mathcal{R}$*

$$sv \models^{sym} R(\alpha_1, \dots, \alpha_n) \text{ iff } v \models_{\mathcal{D}} f(R(\alpha_1, \dots, \alpha_n)).$$

**PROOF.** Let us suppose that  $sv \models^{sym} R(\alpha_1, \dots, \alpha_n)$ . By definition,  $sv \models^{sym} R(\alpha_1, \dots, \alpha_n)$  if for every  $\mathcal{D}$ -valuation  $v'$  over the set of terms within  $sv$  such that  $v' \models_{\mathcal{D}} sv$  it holds that  $v' \models_{\mathcal{D}} f(R(\alpha_1, \dots, \alpha_n))$ . Therefore, we have immediately that  $v \models_{\mathcal{D}} f(R(\alpha_1, \dots, \alpha_n))$ . The converse is an immediate consequence of Lemma 5.1.  $\square$

**LEMMA 5.3.** *Let  $\phi$  be a  $CLTLB(\mathcal{D})$  formula and  $\sigma_k$  be a finite sequence of  $\mathcal{D}$ -valuations. Then, there exists an unique locally consistent sequence  $\rho \in SV(\phi)^{k+1}$  such that  $\sigma_k, i \models \rho(i)$ , for all  $i \in [0, k]$ .*

**PROOF.** By Lemma 5.1 we have that, for all  $i \in [0, k]$ , the assignment of variables defined by  $\sigma_k$  is such that  $\sigma_k, i \models \rho(i)$  and  $\rho(i)$  is unique. By Corollary 5.2, values in  $\sigma_k$  from position  $i$  satisfy a relation  $R$  at position  $i$  if, and only if,  $R$  belongs to symbolic valuation  $\rho(i)$  at position  $i$ , i.e.,  $\rho(i) \models^{sym} R$  iff  $\sigma_k, i \models f(R)$ . We have to show local consistency between two adjacent symbolic valuation. Let us consider  $\rho(i)$  and  $\rho(i+1)$ . It holds that  $R(X^{i_1}x_1, \dots, X^{i_n}x_n) \in \rho(i)$  and  $R(X^{i_1-1}x_1, \dots, X^{i_n-1}x_n) \in \rho(i+1)$  by the uniqueness of values in  $\sigma_k$ . In fact, arithmetic term  $X^{i_j}x_j$  evaluated from position  $i$  has the same value as  $X^{i_j-1}x_j$  evaluated from position  $i+1$ .  $\square$

It is worth noting that by definition of  $|PropConstraints|_k$ ,  $|ArithConstraints|_k$  (see Section 4), when  $|\phi|_k$  is satisfiable we obtain a model  $\sigma_k$  that defines a value for all variables in  $[[\phi], k + \lceil \phi \rceil]$  and a unique symbolic model  $\rho \in SV(\phi)^{k+1}$ .

Before presenting the core result of this section, which shows the correctness of the encoding of Section 4, we introduce two further intermediate results; these allow us to build our correctness result on the automata-based construction for CLTLB( $\mathcal{D}$ ) formulae introduced in [Demri and D'Souza 2007].

More precisely, we provide the following two reductions:

- I. CLTLB( $\mathcal{D}, \mathcal{R}_0$ ) formulae can be rewritten into CLTLB( $\mathcal{D}$ ) formulae,
- II. CLTLB( $\mathcal{D}$ ) formulae can be rewritten into CLTLB( $\mathcal{D}$ ) formulae without Y operators.

### Removing 0-ary relations

According to the definition given in Section 2.2, CLTLB( $\mathcal{D}$ ) is the language CLTLB where atomic formulae belong to the language of constraints in  $\mathcal{D}$ , which may contain also 0-ary relations. In this case, atomic formulae are propositions  $p \in \mathcal{R}_0$  or relations over terms  $R(\alpha_1, \dots, \alpha_n)$ . Any positive occurrence of an atomic proposition  $p \in \mathcal{R}_0$  in a CLTLB formula can be replaced by an equality relation of the form  $x_p = 1$ . Then, a formula of CLTLB( $\mathcal{D}, \mathcal{R}_0$ ) can be easily rewritten into a formula of CLTLB( $\mathcal{D}$ ) preserving the equivalence between them (modulo rewriting of propositions in  $\mathcal{R}_0$ ). We define a rewriting function  $r$  over formulae such that  $\sigma, 0 \models \phi$  if, and only if,  $\theta, 0 \models r(\phi) \wedge \psi$  where  $\theta$  is the same as  $\sigma$  except for new fresh variables  $x_p$  representing atomic propositions and  $\psi$  is a formula restricting values of variables  $x_p$  in  $\{0, 1\}$ .

Let us suppose  $\mathcal{R}_0 = \{p_1, \dots, p_n\}$  to be a finite ordered set of propositions,  $\phi$  a CLTLB( $\mathcal{D}, \mathcal{R}_0$ ) formula and  $V$  the set of variables occurring in  $\phi$ . Let us define  $V_{\mathcal{R}_0} = \{x_{p_1}, \dots, x_{p_n}\}$  as the set of variables representing propositions in  $\mathcal{R}_0$  such that  $V \cap V_{\mathcal{R}_0} = \emptyset$ . We define  $r : \text{CLTLB}(\mathcal{D}, \mathcal{R}_0) \rightarrow \text{CLTLB}(\mathcal{D})$  as the function that maps a formula  $\phi$  to a formula  $\phi'$  identical to  $\phi$  except for all occurrences of any proposition  $p$  in  $\phi$  being replaced in  $\phi'$  by the equality  $x_p = 1$ .

Removing propositions is a syntactic rewriting which acts on formulae. We also provide a syntactic rewriting function  $r_{model}$  which acts on models  $\sigma$  of CLTLB( $\mathcal{D}, \mathcal{R}_0$ ) formulae which replaces occurrences of propositions  $p \in \mathcal{R}_0$  with  $x_p \in V_{\mathcal{R}_0}$ . Let  $\theta = r_{model}(\pi, \sigma)$  be a sequence  $(D^{|V|+n})^\omega$  of valuations of variables in  $V \cup V_{\mathcal{R}_0}$ ; i.e.,  $\theta : \mathbb{Z} \times V \cup \{x_{p_1}, \dots, x_{p_n}\} \rightarrow D$  is the rewriting of  $\sigma$  and  $\pi$  defined as follows:

$$\begin{aligned} \theta(i, x) &= \sigma(i, x) \text{ for all } x \in V, \text{ for all } \lfloor \phi \rfloor \leq i \\ \theta(i, x_{p_j}) &= \begin{cases} 1 & p_j \in \pi(i) \\ 0 & p_j \notin \pi(i) \end{cases} \text{ for all } j \in [1, n] \text{ and for all } i \geq 0. \end{aligned}$$

**PROPOSITION 5.4.** *Let  $\phi$  be a CLTLB( $\mathcal{D}, \mathcal{R}_0$ ) formula where  $\mathcal{R}_0 = \{p_1, \dots, p_n\}$ . Then,  $(\pi, \sigma), 0 \models \phi$  if, and only if,*

$$\theta, 0 \models \left( r(\phi) \wedge \mathbf{G} \left( \bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0) \right) \right)$$

where  $\theta = r_{model}(\pi, \sigma)$ . The proof can be found in Appendix 9.1.

### Removing Y operators

Suppose the formula  $\phi$  contains some a.t.t. of the form  $Y^i x$ . Note that in this case we have  $\lfloor \phi \rfloor < 0$ . We define  $p : \text{CLTLB}(\mathcal{D}) \rightarrow \text{CLTLB}(\mathcal{D})$  as the function that maps a formula  $\phi$  to an equisatisfiable formula  $\phi'$  that does not contain any occurrence of the Y operator. The formula  $\phi'$  is identical to  $\phi$  except for all a.t.t.'s of the form  $X^i x$  in  $\phi$  being replaced in  $\phi'$  by  $X^{i-\lfloor \phi \rfloor} x$ , and a.t.t.'s of the form  $Y^i x$  being replaced by  $Y^{i+\lfloor \phi \rfloor} x$ .

Formally, it can happen, in the transformation above, that some indexes of Y operators become negative (e.g., if  $\lfloor \phi \rfloor = -3$ , then  $p(Y^1 x)$  is replaced by  $Y^{-2}$ ). Then we stipulate that  $Y^{-i} = X^i$

(in the previous example,  $Y^{-2}$  becomes  $X^2$ ). As a consequence, given a  $CLTLB(\mathcal{D})$  formula  $\phi$ , it is easy to see that  $Y$  does not occur in  $p(\phi)$ . The equisatisfiability of formulae  $\phi$  and  $p(\phi)$  is guaranteed by moving the origin of  $\phi$  by  $-|\phi|$  instants in the past. Since only  $X$  occurs in  $p(\phi)$ , then models for  $CLTLB(\mathcal{D})$  formulae without  $Y$  are now sequences of  $\mathcal{D}$ -valuations  $\sigma : \mathbb{N} \times V \rightarrow D$ .

**PROPOSITION 5.5.** *Let  $\phi$  be a  $CLTLB(\mathcal{D})$  formula, then  $\sigma, 0 \models \phi \Leftrightarrow \sigma, \lfloor \phi \rfloor \models p(\phi)$ .*

The proof can be found in Appendix 9.2.

Rewriting function  $p$  naturally extends to the set  $SV(\phi)$  of symbolic valuations to define a new set  $SV'(\phi)$  of symbolic valuations  $sv'$  such that all  $\mathcal{D}$ -formulae of  $sv' \in SV'(\phi)$  consist of relations  $R \in \mathcal{R}$  of future terms of the form  $X^i x$  with  $i \geq 0$ . It is worth noting that the following corollaries hold for generic sequences of symbolic valuations  $\rho$ .

**COROLLARY 5.6.** *Let  $\rho$  be a sequence of symbolic valuations such that  $s \leq 0$  is the minimum value  $i$  occurring in terms of the form  $Y^i x$  and  $\sigma$  is a sequence of  $\mathcal{D}$ -valuations. Then,*

$$\sigma, 0 \models \rho \quad \text{iff} \quad \sigma, s \models p(\rho).$$

**PROOF.** The proof is a consequence of the base case in proof of Proposition 5.5 for all atomic formulae  $R(\alpha_1, \dots, \alpha_n)$  in symbolic valuations constituting  $\rho$ .  $\square$

**COROLLARY 5.7.** *Let  $\phi$  be a  $CLTLB(\mathcal{D})$  formula and  $\rho$  be a sequence of symbolic valuations. Then,*

$$\rho, 0 \models^{\text{sym}} \phi \quad \text{iff} \quad p(\rho), 0 \models^{\text{sym}} p(\phi).$$

The proof can be found in Appendix 9.3.

### Correctness

We now have all necessary elements to prove the correctness of our encoding. We start by showing the equivalence of the satisfiability of  $|\phi|_k$  with the existence of ultimately periodic runs of automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$ .

**THEOREM 5.8.** *Let  $\phi \in CLTLB(\mathcal{D})$  with  $\mathbb{N}$  definable in  $\mathcal{D}$  together with the successor relation,  $|\phi|_k$  is satisfiable with respect to  $k \in \mathbb{N}$  if, and only if, there exists an ultimately periodic run  $\rho = \alpha\beta^\omega$  ( $|\alpha\beta| = k + 1$ ) of  $\mathcal{A}_s \times \mathcal{A}_\ell$  accepting symbolic models of  $\phi$ .*

Before proving the theorem we provide the definition of models for QF-EUD formulae. Domain  $D$  and the interpretation of relations are constrained by  $\mathcal{D}$ . A model  $\mathcal{M}$  is a pair  $(D, \mathcal{I})$  where  $\mathcal{I}$  is an interpretation defined as the mapping:

- for all function symbols  $\alpha$  a function associating, for each position of time, an element in domain  $D$ ,  $\mathcal{I}(\alpha) : \mathbb{N} \rightarrow D$ ,
- for all predicate symbols  $\theta$  a function associating, for each position of time, an element in  $\{true, false\}$ ,  $\mathcal{I}(\theta) : \mathbb{N} \rightarrow \{true, false\}$ .

Provided that  $D$  contains a copy of  $\mathbb{N}$ , then, given an interpretation  $\mathcal{I}$ , the values for all formula functions  $\alpha$  and predicate functions  $\theta$  encoding terms and subformulae of a  $CLTLB(\mathcal{D})$  formula are known. Interpretation  $\mathcal{I}$  trivially induces a model  $\sigma_k : \{\lfloor \phi \rfloor, \dots, k + \lceil \phi \rceil\} : V \rightarrow D$ .

In the following proof, we will use the notion of “accepting subrun” of the Büchi automaton obtained by the standard construction of [Vardi and Wolper 1986], in the version of [Demri and D’Souza 2007]. Let  $\phi$  be a  $CLTLB(\mathcal{D})$  formula (without the  $Y$  modality over terms). The closure of  $\phi$ , denoted  $cl(\phi)$ , is the smallest set containing all subformulae of  $\phi$  that is also closed under negation. An *atom*  $\Gamma \subseteq cl(\phi)$  is a subset of formulae of  $cl(\phi)$  that is maximally consistent, i.e., such that, for each formula  $\xi$  in  $\phi$ , either  $\xi \in \Gamma$  or  $\neg\xi \in \Gamma$ . A pair  $(\Gamma_1, \Gamma_2)$  of atoms is *one-step temporally consistent* when:

- for every  $X\psi \in cl(\phi)$ , then  $X\psi \in \Gamma_1 \Leftrightarrow \psi \in \Gamma_2$ ,

- for every  $\mathbf{Y}\psi \in cl(\phi)$ , then  $\mathbf{Y}\psi \in \Gamma_2 \Leftrightarrow \psi \in \Gamma_1$ ,
- if  $\psi_1 \mathbf{U}\psi_2 \in \Gamma_1$ , then  $\psi_2 \in \Gamma_1$  or  $(\psi_1 \in \Gamma_1 \text{ and } \psi_1 \mathbf{U}\psi_2 \in \Gamma_2)$ ,
- if  $\psi_1 \mathbf{S}\psi_2 \in \Gamma_2$ , then  $\psi_2 \in \Gamma_2$  or  $(\psi_1 \in \Gamma_2 \text{ and } \psi_1 \mathbf{S}\psi_2 \in \Gamma_1)$ .

The automaton  $\mathcal{A}_s = (SV(\phi), Q, Q_0, \eta, F)$  is then defined as follows:

- $Q$  is the set of atoms;
- $Q_0 = \{\Gamma \in Q : \phi \in \Gamma, \mathbf{Y}\psi \notin \Gamma \text{ for all } \psi \in cl(\phi), \psi_1 \mathbf{S}\psi_2 \in \Gamma \text{ iff } \psi_2 \in \Gamma\}$ ;
- $\Gamma_1 \xrightarrow{sv} \Gamma_2 \in \eta$  iff
  - $\Gamma_1 \models^{sym} sv$
  - $(\Gamma_1, \Gamma_2)$  is one-step consistent;
- $F = \{F_1, \dots, F_p\}$ , where  $F_i = \{\Gamma \in Q \mid \psi_i \mathbf{U}\zeta_i \notin \Gamma \text{ or } \zeta_i \in \Gamma\}$  and  $\{\psi_1 \mathbf{U}\zeta_1, \dots, \psi_p \mathbf{U}\zeta_p\}$  is the set of Until formulae occurring in  $cl(\phi)$ .

An *accepting subrun* for  $\theta \in cl(\phi)$  is a finite sequence of atoms  $\Gamma_1, \Gamma_2, \dots, \Gamma_m$  such that:

- $\Gamma_1 \xrightarrow{sv_1} \Gamma_2 \xrightarrow{sv_2} \dots \xrightarrow{sv_{m-1}} \Gamma_m$ ;
- $\theta \in \Gamma_1$ , if  $\theta = \mathbf{X}\psi$  or  $\theta = \psi_1 \mathbf{U}\psi_2$ ;
- $\theta \in \Gamma_m$ , if  $\theta = \mathbf{Y}\psi$  or  $\theta = \psi_1 \mathbf{S}\psi_2$ ;
- $\Gamma_m \in F$  if  $\theta = \psi_1 \mathbf{U}\psi_2$

Moreover, observe that  $m = 2$  if  $\theta = \mathbf{X}\psi$  or  $\theta = \mathbf{Y}\psi$ .

PROOF. We prove the lemma by showing that formula  $|\phi|_k$  represents accepting runs of the product automaton of  $\mathcal{A}_s$ , the symbolic Büchi automaton of  $\phi$ , and  $\mathcal{A}_\ell$ . If formula  $|\phi|_k$  is satisfiable, we prove that, given a subformula  $\theta \in cl(\phi)$ , if  $\theta$  holds then there exists a control state of  $\mathcal{A}_s \times \mathcal{A}_\ell$ , defined by an atom  $\Gamma$  such that  $\theta \in \Gamma$  and which is visited by some initialized run. Observe that the encoding of  $|\phi|_k$  defines precisely the truth value of all subformulae  $\theta$  of  $\phi$  in instants  $i \in [0, k]$ . Then, if  $|\phi|_k$  is satisfiable, given an  $i \in [0, k]$ , the set of all subformulae

$$\Gamma_i = \{\varphi \in cl(\phi) \mid \text{if } \theta \text{ holds in } i \text{ then } \varphi = \theta, \text{ else } \varphi = \neg\theta\}$$

is an atom of automaton  $\mathcal{A}_s$ . Let us suppose  $\mathbf{loop} \in [1, k]$ . The sequence of sets  $\Gamma_i$  for  $0 \leq i \leq k$  is an ultimately periodic sequence of atoms of  $\mathcal{A}_s$  due to the satisfiability of formulae  $|LastStateConstraints|_k$  and  $|LoopConstraints|_k$ . We write  $\Gamma|_A$  to denote the projection of  $\mathcal{D}$ -constraints in  $\Gamma$  on symbols of the set  $A$ ; e.g., if  $A = \{R_1, R_2\}$  then  $\{R_1(x, y), R_2(Xx, Yx), \theta_1, \theta_2\}|_A = \{R_1(x, y), R_2(Xx, Yx)\}$ . The sequence of atoms is

$$\gamma = \Gamma_0 \dots \Gamma_{\ell-1} (\Gamma_\ell \dots, \Gamma_k)^\omega$$

and such that  $\Gamma_{\ell-1}|_{\mathcal{R}}$  is equal to the set of relations of  $\Gamma_k|_{\mathcal{R}}$  by formulae in  $|LoopConstraints|_k$ . Moreover, by  $|LastStateConstraints|_k$  we have  $\Gamma_{k+1}|_{cl(\phi)} = \Gamma_\ell|_{cl(\phi)}$ .

We now show that we can obtain, from an interpretation  $\mathcal{I}$  satisfying  $|\phi|_k$ , an ultimately periodic run which does not contain the  $\mathbf{Y}$  modality over terms and which is a run of automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$ . By Proposition 5.4, the truth value of  $\phi$  is preserved, modulo a rewriting defined by  $r$ , by replacing 0-ary relations (atomic propositions) in interpretation  $\mathcal{I}$  by formulae of the form either  $x_p = 1$  or  $x_p = 0$ , where  $p \in \mathcal{R}_0$ . Interpretation  $\mathcal{I}$  can be completed in all positions  $\lfloor \phi \rfloor \leq i < 0$ , for all variables  $x_p \in V_{\mathcal{R}_0}$ , by assigning an arbitrary value in  $\{0, 1\}$ . In fact, truth values of atomic propositions in  $\mathcal{R}_0$  before 0 do not affect the evaluation of formula  $\phi$ , by definition of relation  $\models$ . Then, we are allowed to choose any value to complete the model of variables before 0. Therefore, we obtain a model  $\mathcal{M}' = (D, \mathcal{I}')$  which is a  $k$ -bounded model for formula  $r(\phi)$  where atomic propositions are replaced by equality relations. By Lemma 5.3, from  $\sigma_k$ , induced by  $\mathcal{I}'$ , we have a unique locally consistent sequence of symbolic valuations  $\rho$  such that  $\sigma_k, 0 \models_k \rho$ . The sequence  $\rho$  of symbolic valuations is such that all atomic propositions  $p \in \mathcal{R}_0$  are replaced by equality relations of the form  $x_p = 1$ , if  $p \in sv$ , otherwise  $x_p = 0$ . Observe that, in order to simplify the notation, we write  $\sigma_k, 0 \models_k \rho$  even when using rewriting  $r$  (and the corresponding  $r_{model}$ ), instead of the more

precise, but lengthier,  $r_{model}(\sigma_k), 0 \models_k r(\rho)$ . Formula  $|LoopConstraints|_k$  witnesses ultimately periodic sequences of symbolic valuations  $\rho$  because it is defined over the set of relations in  $\mathcal{R}$  and all terms of the set  $terms(\phi)$ :

$$\rho = sv_0 \dots sv_{\ell-1} (sv_{\ell} \dots sv_k)^\omega$$

such that  $sv_{\ell-1} = sv_k$ . By Corollary 5.6, model  $\sigma_k$  is such that  $\sigma_k, [\phi] \models p(\rho)$  where:

$$p(\rho) = p(sv_0)_{[\phi]} \dots p(sv_{\ell-1})_{\ell-1+[\phi]} (p(sv_{\ell})_{\ell+[\phi]} \dots p(sv_k)_{k+[\phi]})^\omega.$$

and  $p(\rho), 0 \models^{sym} p(\phi)$ , by Corollary 5.7. Position  $[\phi]$  is the origin of sequence  $p(\rho)$  of symbolic valuations which do not contain the past modality  $Y$  over terms. By Proposition 5.5, we can shift the sequence of atoms  $\Gamma_i$  of  $[\phi]$  positions backward to the past. The new sequence

$$p(\gamma) = p(\Gamma_0)_{[\phi]} \dots p(\Gamma_{\ell-1})_{\ell-1+[\phi]} (p(\Gamma_{\ell})_{\ell+[\phi]} \dots p(\Gamma_k)_{k+[\phi]})^\omega$$

is such that for each  $\Gamma_i$ ,  $\sigma_k, 0 \models \Gamma_i \Leftrightarrow \sigma_k, [\phi] \models p(\Gamma_i)$ , where  $\sigma_k, i \models \Gamma$  if, and only if,  $\sigma_k, i \models \theta$  for all  $\theta \in \Gamma$  and  $p(\Gamma) = \{p(\theta) \mid \theta \in \Gamma\}$ .

Consequently, if the encoding  $[\phi_k]$  of  $\phi$  is satisfiable with a finite model  $\mathcal{M}$ , then  $p(\gamma)$  is an accepting run of automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$  of formula  $p(\phi)$  which recognizes sequences of symbolic valuations  $p(\rho)$  such that  $p(\phi) \in p(\Gamma_0)_{[\phi]}, p(\rho), 0 \models^{sym} p(\phi)$  and  $\sigma_k, [\phi] \models_k p(\rho')$ , where  $\rho' = sv_0 \dots sv_{\ell-1} sv_{\ell} \dots sv_k$ . By Propositions 5.5 and 5.4 from  $p(\gamma)$  and  $p(\rho)$  we can build an accepting run  $\Gamma_0 \dots \Gamma_{\ell-1} (\Gamma_{\ell} \dots \Gamma_k)^\omega$  of automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$  of  $\phi$  recognizing  $\rho$ . In fact, from  $p(\rho)$  we can obtain  $\rho$  and from  $p(\gamma)$  we can obtain  $\gamma$  by means of the inverse rewriting  $p^{-1}$ . Moreover, atomic propositions can also be restored by means of the inverse of  $r$ .

Therefore, without loss of generality, let us suppose  $\phi$  to be a formula where neither the  $Y$  modality on terms, nor atomic propositions occur.

Now, we provide the second step of the proof. More precisely, by induction on the structure of formula  $\phi$ , we prove that for each position  $0 \leq i \leq k$ , if  $\theta_i$  holds in the finite model, i.e.,  $\theta_i = true$ , then there exists a control state  $\Gamma$  in automaton  $\mathcal{A}_s$  such that  $\theta_i \in \Gamma$ .

Recall that the encoding is such that:

- $\Gamma_{loop|cl(\phi)} = \Gamma_{k+1|cl(\phi)}$ . The encoding of subformulae  $(|LastStateConstraints|_k)$  is such that if there exists a loop then for all subformulae  $\theta_{k+1} \Leftrightarrow \theta_{loop}$ .
- $sv_{loop-1} = sv_k$ .
- No periodic constraints are defined for the arithmetic model  $\sigma_k$ .

Now, we prove by structural induction on  $\phi$  that for  $0 \leq i \leq k$ ,  $\theta_i$  holds if, and only if, there exists a sequence of atoms  $\Gamma_i, \dots, \Gamma_m$  defining an accepting subrun of  $\mathcal{A}_s$  for the formula  $\theta$ .

The **base case** is given on relations formulae  $\theta = R(\alpha_1, \dots, \alpha_n)$ . If  $\theta_i$  holds, with  $0 \leq i \leq k$ , then there exists a symbolic valuation  $sv$  such that  $sv_i$  is satisfiable at the position  $i$  and also  $\theta \in sv_i$ ; i.e.,  $sv \models^{sym} \theta$ , as required in the rule defining the transition relation of automaton  $\mathcal{A}_s$ . Also, there

are two control states  $sv'$  and  $sv''$  such that  $sv_{i-1} = sv' \xrightarrow{sv'} sv$  and  $sv \xrightarrow{sv} sv'' = sv_{i+1}$  for  $i \geq 1$ . This follows from the consistency of the encoding as considered in the proof of Lemma 5.3. The set of subformulae  $\Gamma_i$  defines an atom constituting a control state of the automaton  $\mathcal{A}_s$  such that  $sv = \Gamma_i \cap SV(\phi)$ . Hence,  $\Gamma_i$  of  $\mathcal{A}_s$  and  $sv_i$  of  $\mathcal{A}_\ell$  are such that  $\theta \in \Gamma_i, \Gamma_i \xrightarrow{sv} \Gamma''$ , for some atom  $\Gamma''$ , and  $sv \xrightarrow{sv} sv''$ , for some  $sv''$  which is locally consistent with  $sv$ .

Then, we proceed with the **inductive step** by considering all subformulae in  $cl(\phi)$ . We show that for each  $\theta \in cl(\phi)$  it is possible to build an accepting subrun of  $\mathcal{A}_s$ . Therefore, inductively, we have  $\phi \in \Gamma_0$  (or, equivalently,  $p(\phi) \in p(\Gamma_0)$ )

- If  $\theta = \mathbf{X}\psi$  then for  $0 \leq i < k$ ,  $\theta_i \Leftrightarrow \psi_{i+1}$ . By inductive hypothesis, we have that  $\psi \in \Gamma_{i+1}$ . Then, by  $|TempConstraints|_k$ ,  $\mathbf{X}\psi \in \Gamma_i$ , that is  $\theta \in \Gamma_i$ . Moreover, there are two locally consistent symbolic valuations  $sv, sv' \in SV(\phi)$  such that  $sv_i = sv \xrightarrow{sv} sv' = sv_{i+1}$  which is a subrun of



- $\mathcal{A}_\ell$ . It follows that  $\Gamma_i \xrightarrow{sv} \Gamma_{i+1}$  such that  $sv \in \Gamma_i \cap SV(\phi)$  is a subrun of  $\mathcal{A}_s$  for which  $\mathbf{X}\psi \in \Gamma_i$  iff  $\psi \in \Gamma_{i+1}$  according to the definition of one-step consistent atoms. If  $i = k$ , the same arguments as above can be used provided that we consider  $\theta_i \Leftrightarrow \psi_{loop}$  and  $\psi \in \Gamma_{loop}$ .
- If  $\theta = \mathbf{Y}\psi$  then for  $1 \leq i \leq k$ ,  $\theta_i \Leftrightarrow \psi_{i-1}$ ; in addition  $\theta_0 \Leftrightarrow \perp$ . By inductive hypothesis, we have that  $\psi \in \Gamma_{i-1}$ . Then, by  $|TempConstraints|_k$ ,  $\mathbf{X}\psi \in \Gamma_i$ , that is  $\theta \in \Gamma_i$ . Moreover, there are two locally consistent symbolic valuations  $sv, sv' \in SV(\phi)$  such that  $sv_{i-1} = sv \xrightarrow{sv} sv' = sv_i$  which is a subrun of  $\mathcal{A}_\ell$ . It follows that  $\Gamma_{i-1} \xrightarrow{sv} \Gamma_i$  such that  $sv \in \Gamma_{i-1} \cap SV(\phi)$  is a subrun of  $\mathcal{A}_s$  for which  $\mathbf{Y}\psi \in \Gamma_i$  iff  $\psi \in \Gamma_{i-1}$  according to the definition of one-step consistent atoms.  $\theta_0 \Leftrightarrow \perp$  defines the initial atom  $\Gamma_0$  according to the rule defining initial states of Vardi-Wolper automata.
  - If  $\theta = \psi\mathbf{U}\zeta$  then for  $0 \leq i \leq k$ ,  $\theta_i \Leftrightarrow \zeta_i \vee (\psi_i \wedge (\psi\mathbf{U}\zeta)_{i+1})$ . Two cases have to be considered.
    - $\zeta_j$  holds for some  $i \leq j \leq k$ . By inductive hypothesis,  $\zeta \in \Gamma_j$ ,  $\psi \in \Gamma_i \dots \Gamma_{j-1}$  and  $\theta \in \Gamma_{i+1} \dots \Gamma_j$ . By  $|TempConstraints|_k$ ,  $\theta = \psi\mathbf{U}\zeta \in \Gamma_i$ . There is a sequence of locally consistent symbolic valuations  $sv_i \dots sv_j$  such that  $sv_i \xrightarrow{sv_i} \dots \xrightarrow{sv_{j-1}} sv_j$  which is a subrun of  $\mathcal{A}_\ell$ . It follows that  $\Gamma_i \xrightarrow{sv_i} \dots \xrightarrow{sv_{j-1}} \Gamma_j$  where, for all  $i \leq z \leq j$ ,  $sv_z = \Gamma_z \cap SV(\phi)$ , is a subrun of  $\mathcal{A}_s$  for which if  $\psi_1\mathbf{U}\psi_2 \in \Gamma_i$  then  $\psi_2 \in \Gamma_i$  or  $(\psi_1 \in \Gamma_i$  and  $\psi_1\mathbf{U}\psi_2 \in \Gamma_{i+1})$  according to the definitions of one-step consistent atoms.
    - Otherwise,  $\zeta_j$  holds for some  $j, loop \leq j \leq k$ ; therefore, from formula:  $\theta_k \Rightarrow \ell \leq j \leq k \wedge \zeta_j$  in  $|TemporalConstraints|_k$  and by  $|LastStateConstraints|_k$ , which enforces  $\theta_{k+1} \Leftrightarrow \theta_{loop}$ ,  $\theta$  holds in all positions  $i+1 \leq z \leq k+1$  and also in  $loop \leq z \leq j$ . Similarly to the previous case, we can build the accepting sequence for  $\theta$  by considering the positions in the finite model where  $\theta$  is satisfied. In particular, we consider  $\Gamma_i \dots \Gamma_k \Gamma_{loop} \dots \Gamma_j$  the accepting subrun of  $\theta$  in  $\mathcal{A}_s$ . In fact, by inductive hypothesis,  $\zeta \in \Gamma_j$  and  $\psi \in \Gamma_i \dots \Gamma_k \Gamma_{loop} \dots \Gamma_{j-1}$ . Then,  $\psi\mathbf{U}\zeta \in \Gamma_i$ . The sequence of locally consistent symbolic valuations is  $sv_i \dots sv_k sv_{loop} \dots sv_j$  s.t.  $sv_i \xrightarrow{sv_i} \dots \xrightarrow{sv_k} sv_{loop} \xrightarrow{sv_{loop}} \dots \xrightarrow{sv_{j-1}} sv_j$  is a subrun of  $\mathcal{A}_\ell$ .
  - If  $\theta = \psi\mathbf{S}\zeta$  then for  $1 \leq i \leq k+1$ ,  $\theta_i \Leftrightarrow \zeta_i \vee (\psi_i \wedge (\psi\mathbf{S}\zeta)_{i-1})$  and also  $(\psi\mathbf{S}\zeta)_0 \Leftrightarrow \zeta_0$ . Then,  $\zeta_j$  holds for  $0 \leq j \leq i$ . By inductive hypothesis,  $\zeta \in \Gamma_j$ ,  $\psi \in \Gamma_j \dots \Gamma_i$  and  $\theta \in \Gamma_j \dots \Gamma_{i-1}$ . By  $|TempConstraints|_k$ ,  $\theta = \psi\mathbf{S}\zeta \in \Gamma_i$ . There is a sequence of locally consistent symbolic valuations  $sv_j \dots sv_i$  such that  $sv_j \xrightarrow{sv_j} \dots \xrightarrow{sv_{i-1}} sv_i$  which is a subrun of  $\mathcal{A}_\ell$ . It follows that  $\Gamma_j \xrightarrow{sv_j} \dots \xrightarrow{sv_{i-1}} \Gamma_i$  where, for all  $j \leq z \leq i$ ,  $sv_z = \Gamma_z \cap SV(\phi)$ , is a subrun of  $\mathcal{A}_s$  for which if  $\psi_1\mathbf{S}\psi_2 \in \Gamma_i$  then  $\psi_2 \in \Gamma_i$  or  $(\psi_1 \in \Gamma_i$  and  $\psi_1\mathbf{U}\psi_2 \in \Gamma_{i-1})$  according to the definitions of one-step consistent atoms.  $(\psi\mathbf{S}\zeta)_0 \Leftrightarrow \zeta_0$  defines initial atoms  $\Gamma_0$  according to the rule defining initial states of Vardi-Wolper automata.
  - If  $\theta = \psi\mathbf{R}\zeta$  then for  $0 \leq i \leq k$ ,  $\theta_i \Leftrightarrow \zeta_i \wedge (\psi_i \vee (\psi\mathbf{R}\zeta)_{i+1})$ . This case can be reduced to the analysis of a subformula containing  $\mathbf{U}$ . In fact, by duality of  $\mathbf{U}$  and  $\mathbf{R}$ ,  $\neg\theta = \neg(\psi\mathbf{R}\zeta) = \neg\psi\mathbf{U}\neg\zeta$ . Therefore, subruns accepting  $\neg\theta$  are accepting subruns for  $\neg\psi\mathbf{U}\neg\zeta$ ; then, subruns accepting  $\theta$  are all non accepting subruns for  $\neg\psi\mathbf{U}\neg\zeta$ .
  - If  $\theta = \psi\mathbf{T}\zeta$  then for  $1 \leq i \leq k+1$ ,  $\theta_i \Leftrightarrow \psi_i \wedge (\psi_i \vee (\psi\mathbf{T}\zeta)_{i-1})$  and  $(\psi\mathbf{T}\zeta)_0 \Leftrightarrow \zeta_0$ . This case can be reduced to the analysis of a subformula containing  $\mathbf{S}$ . In fact, by duality of  $\mathbf{S}$  and  $\mathbf{T}$ ,  $\neg\theta = \neg(\psi\mathbf{T}\zeta) = \neg\psi\mathbf{S}\neg\zeta$ . Therefore, subruns accepting  $\neg\theta$  are accepting subruns for  $\neg\psi\mathbf{S}\neg\zeta$ ; then, subruns accepting  $\theta$  are all non accepting subruns for  $\neg\psi\mathbf{S}\neg\zeta$ .

The case for  $loop \notin [1, k]$  can be derived from the previous analysis. The sequence of atoms  $\Gamma_0 \dots \Gamma_k$  is a finite run of  $\mathcal{A}_s$  and sequence  $sv_0 \dots sv_k$  of symbolic valuations is a finite word of locally consistent symbolic valuations. If  $\phi$  can be satisfied by  $sv_0 \dots sv_k$  then the evaluation of  $\phi$  does not depend on truth values of its subformulae from position  $k+1$  upwards. Prefix  $sv_0 \dots sv_k$  can be completed by any sequence in  $SV(\phi)^\omega$ . In this case,  $|LastStateConstraints|_k$  enforces  $\perp$  at position  $k+1$  by constraining all subformulae in  $cl(\phi)$  to  $\perp$ . Consequently,  $|TempConstraints|_k$

for future formulae  $\mathbf{X}\phi$ ,  $\phi\mathbf{U}\psi$  and  $\phi\mathbf{R}\psi$  are:

$\theta$	$k$
$\mathbf{X}\phi$	$\theta(i) \Leftrightarrow \perp$
$\phi\mathbf{U}\psi$	$\theta(i) \Leftrightarrow \psi(i)$
$\phi\mathbf{R}\psi$	$\theta(i) \Leftrightarrow \psi(i) \wedge \phi(i)$

In this case, the value at position  $k$  of formulae of the form  $\phi\mathbf{U}\psi$  and  $\phi\mathbf{R}\psi$  depends only on the truth value of  $\psi$  and  $\phi$  at position  $k$ . In fact  $\phi\mathbf{U}\psi$  in  $k$  if, and only if, formula  $\psi$  holds at the same position  $k$ ; whereas  $\phi\mathbf{R}\psi$  in  $k$  if, and only if, formula  $\psi$  and  $\phi$  hold at the same position  $k$ . The analysis of the inductive step is the same as the previous case for  $\text{loop} \in [1, k]$  except for:

- formula  $\theta = \mathbf{X}\phi$  is considered only for positions  $0 \leq i < k$ ;
- the second case of  $\theta = \phi\mathbf{U}\zeta$ , i.e.,  $\zeta_i$  with  $\text{loop} \leq i \leq k$ , is no longer needed. In fact, if a formula  $\theta = \phi\mathbf{U}\zeta$  holds at position  $i$ , then  $\zeta_j$  holds for  $i \leq j \leq k$  over a finite subrun.

To conclude the first part of the proof, if  $|\phi|_k$  is satisfiable then  $\phi$  holds at time 0, hence  $\phi \in \Gamma_0$  ( $p(\phi) \in p(\Gamma_0)$ ), where  $\Gamma_0$  is an initial state of  $\mathcal{A}_s$  because it satisfies  $|\text{TempConstraints}|_k$  at time 0; then, the sequence of atoms  $\gamma$  is a periodic accepting run of  $\mathcal{A}_s$  for  $\phi$ , accepting an ultimately periodic symbolic model  $\rho$ .

Let us now prove that if there is a run in  $\mathcal{A}_s \times \mathcal{A}_\ell$  accepting  $p(\phi)$ , then formula  $|\phi|_k$  is satisfiable (again we assume the rewriting induced by  $r$ ). Let us suppose there exists an ultimately periodic symbolic model of length  $k + 1$  which is accepted by  $\mathcal{A}_s \times \mathcal{A}_\ell$ . It is a locally consistent sequence of symbolic valuations,  $\rho = \alpha\beta^\omega$  of the form:

$$\rho = sv_0 \dots sv_{\text{loop}-1} (sv_{\text{loop}} \dots sv_k)^\omega$$

such that  $\rho \in \mathcal{L}(\mathcal{A}_s \times \mathcal{A}_\ell)$  and which is recognized by a periodic run of  $\mathcal{A}_s \times \mathcal{A}_\ell$  of the form<sup>2</sup>:

$$v = (\Gamma_0, sv_0) \dots (\Gamma_{\text{loop}-1}, sv_{\text{loop}-1}) ((\Gamma_{\text{loop}}, sv_{\text{loop}}) \dots (\Gamma_k, sv_k))^\omega.$$

For each subformula  $\psi_i \mathbf{U} \zeta_i$  occurring in  $\phi$ , subrun  $(\Gamma_{\text{loop}-1}, sv_{\text{loop}-1}) (\Gamma_{\text{loop}}, sv_{\text{loop}}) \dots (\Gamma_k, sv_k)$  visits control states of the set  $F_i$ , thus witnessing the acceptance condition of  $\mathcal{A}_s$ . From  $v$  we build run  $\gamma$  of  $\mathcal{A}_s$ :

$$\gamma = \Gamma_0 \dots \Gamma_{\text{loop}-1} (\Gamma_{\text{loop}} \dots \Gamma_k)^\omega.$$

In particular,  $\rho$  is defined by the projection on the alphabet of  $SV(p(\phi))$  of the subformulae occurring in every  $\Gamma_i$ , for  $0 \leq i \leq k$ . Sequence  $\rho$  and its accepting run  $\gamma$  can be translated by means of  $p^{-1}$  in order to obtain a symbolic model for  $\phi$ . In particular, because  $\rho, 0 \models^{\text{sym}} p(\phi)$  then we obtain  $p^{-1}(\rho), 0 \models^{\text{sym}} \phi$ . Similarly, by shifting all formulae in atoms of  $\gamma$ , we obtain an accepting run  $p^{-1}(\gamma)$  for  $\phi$ . The model for  $|\phi|_k$  is given by the truth value of all the subformulae in  $p^{-1}(\Gamma_i)$  and the values of variables occurring in  $\phi$  defining  $\sigma_k$  can be defined as explained later. In particular, we need to complete interpretation  $\mathcal{I}$  for uninterpreted predicate and functions formulae: given a position  $0 \leq i \leq k$ , for all subformulae  $\theta \in cl(\phi)$  we define

- $\mathcal{I}(\theta)(i) = \text{true}$  iff  $\theta \in p^{-1}(\Gamma_i)$ ,
- $\mathcal{I}(\theta)(i) = \text{false}$  iff  $\neg\theta \in p^{-1}(\Gamma_i)$ .

The truth value of subformulae  $\psi\mathbf{R}\zeta$  and  $\psi\mathbf{T}\zeta$  is derived by duality:  $\neg\psi\mathbf{R}\zeta = \neg\psi\mathbf{U}\neg\zeta$  and  $\neg\psi\mathbf{T}\zeta = \neg\psi\mathbf{S}\neg\zeta$ . To complete the interpretation of subformulae at position  $k + 1$  we can use values from  $\text{loop}$ :  $\mathcal{I}(\theta)(k + 1) = \mathcal{I}(\theta)(\text{loop})$ . Observe that by taking truth values of subformulae  $\theta \in cl(\phi)$  from atoms  $p^{-1}(\Gamma_i)$ ,  $|\text{propConstraints}|_k$  are trivially satisfied (atoms are defined by using the same Boolean closure in  $|\text{propConstraints}|_k$ ). The sequence  $\rho$  of symbolic valuations is consistent

<sup>2</sup>For reasons of clarity, we avoid some details of product automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$ , which are however inessential in the proof.

and all the a.t.t.'s in the encoding of  $|\phi|_k$  can be uniquely defined by considering at each position  $i$  a symbolic valuation  $p^{-1}(sv_i)$ . Consider the sequence  $\rho' = sv_0 \dots sv_{loop-1}(sv_{loop} \dots sv_k)sv_{loop}$ . The model  $\sigma_k(i, x)$  for each variable  $x \in V$  and for  $0 \leq i \leq k + 1 + \lceil \phi \rceil$  is defined by an edge-respecting assignment of values in  $D$  for the graph  $G_{p^{-1}(\rho')}$  according to what is suggested in [Demri and D'Souza 2007, Lemma 5.2]. All a.t.t.'s  $\alpha_i$  are uniquely defined by considering the values of variables in  $\sigma_k$ . We define  $\mathcal{I}(\alpha)$ , with  $\alpha = X^j x$  and  $1 \leq j \leq \lceil \phi \rceil$  or  $\alpha = Y^j x$  and  $1 \leq j \leq -\lfloor \phi \rfloor$ :

$$\mathcal{I}(\alpha)(i) = \sigma_k(i + |\alpha|, x)$$

for all  $0 \leq i \leq k + 1$ . Then, formulae  $|ArithConstraints|_k$  are satisfied. Since run  $v$  is ultimately periodic, then control state  $(\Gamma_{loop}, sv_{loop})$  is visited at position  $k + 1$ . It witnesses the satisfaction of  $|LastStateConstraints|_k$  formulae, which prescribe that  $\theta_{k+1} \Leftrightarrow \theta_{loop}$  for all  $\theta \in cl(\phi)$ . Finally, let us consider  $|Eventually|_k$  formulae. If subformula  $\varphi = \psi U \zeta$  belongs to the atom  $\Gamma_k$ , then there exists a position  $j \geq k$  such that  $\zeta_j$  holds. Since the model is periodic then  $k \leq j \leq 2k$ , i.e.,  $j$  is a position in  $\ell \leq j_\zeta \leq k$ . Moreover, if  $\neg(\psi R \zeta) = \neg\psi U \neg\zeta$  belongs to  $\Gamma_k$  then there exists a position  $j \geq k$  such that  $\neg\zeta_j$  holds. As in the previous case  $\ell \leq j_\zeta \leq k$ . Hence, the  $|Eventually|_k$  formulae are satisfied. The initial atom  $\Gamma_0$  is such that  $\mathbf{Y}\varphi \notin \Gamma_0$  and if  $\psi S \zeta \in \Gamma_0$  then  $\zeta \in \Gamma_0$ , which witnesses the encoding of subformulae  $\mathbf{Y}\psi$  and  $\psi S \zeta$  at 0, i.e.,  $\theta_0 \Leftrightarrow \perp$  and  $\theta_0 \Leftrightarrow \zeta_0$ , respectively.  $\square$

The next theorem draws a link between  $k$ -satisfiability and the existence of an ultimately periodic run in automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$ .

**THEOREM 5.9.** *Let  $\phi \in CLTLB(\mathcal{D})$  with  $\mathbb{N}$  definable in  $\mathcal{D}$  together with the successor relation. Formula  $\phi$  is  $k$ -satisfiable with respect to  $k \in \mathbb{N}$  if, and only if, there exists an ultimately periodic run  $\rho = \alpha\beta^\omega$  of  $\mathcal{A}_s \times \mathcal{A}_\ell$ , with  $|\alpha\beta| = k + 1$ , accepting symbolic models of  $\phi$ .*

**PROOF.** By definition, if  $\phi$  is  $k$ -satisfiable, then there is an ultimately periodic symbolic model  $\rho = \alpha\beta^\omega$  such that  $\rho, 0 \models \phi$ . By Lemma 5.3,  $\rho$  is locally consistent because there exists a  $k$ -bounded model  $\sigma_k$  such that  $\sigma_k \models_k \alpha\beta$ . Therefore,  $\rho \in \mathcal{L}(\mathcal{A}_s \times \mathcal{A}_\ell)$ .

Conversely, if the language of  $\mathcal{A}_s \times \mathcal{A}_\ell$  is not empty, then the automaton accepts also ultimately periodic symbolic models  $\alpha\beta^\omega$  over the alphabet  $SV(\phi)$ , whose prefix has the form  $\alpha s\beta' s$ , where  $\beta = s\beta'$ . Since sequence  $\alpha\beta$  is a finite prefix of length  $k + 1$  of symbolic model  $\alpha\beta^\omega$ , which admits an arithmetic model, then  $\alpha\beta$  admits a  $k$ -bounded model  $\sigma_k$  defined by an edge-respecting labeling of the graph  $G_{\alpha\beta}$ .  $\square$

We can prove the main equivalence result which draws the connection between the encoding and the  $k$ -satisfiability problem.

**THEOREM 5.10.** *Let  $\phi \in CLTLB(\mathcal{D})$  with  $\mathbb{N}$  definable in  $\mathcal{D}$  together with the successor relation,  $\phi$  is  $k$ -satisfiable with respect to  $k \in \mathbb{N}$  if, and only if,  $|\phi|_k$  is satisfiable.*

**PROOF.** It is a direct consequence of Theorems 5.8 and 5.9.  $\square$

As explained in Section 2.4, each automaton involved in the definition of  $\mathcal{A}_\phi$  has the function of “filtering” sequences of symbolic valuations so that 1) they are locally consistent, 2) they satisfy an LTL property and 3) they admit a (arithmetic) model. As mentioned in Section 2, for constraint systems that have the completion property local consistency is a sufficient and necessary condition for admitting a model. For these constraint systems  $\mathcal{A}_\phi$  is exactly automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$ , and from Proposition 2.5 and Theorem 5.10 we obtain the following result.

**PROPOSITION 5.11.** *Let  $\phi \in CLTLB(\mathcal{D})$  with  $\mathbb{N}$  definable in  $\mathcal{D}$  together with the successor relation,  $\phi$  is  $k$ -satisfiable with respect to  $k \in \mathbb{N}$  if, and only if,  $\phi$  has an ultimately periodic model  $\alpha\beta^\omega$  with  $|\alpha\beta| = k + 1$ .*

PROOF. Formula  $\phi$  is  $k$ -satisfiable if, and only if,  $|\phi|_k$  is satisfiable. Then, bounded model  $\sigma_k$  can be extended to (an infinite model)  $\sigma$ , from  $k$  forward, by iterating infinitely many times the suffix  $sv_{loop} \dots sv_k$  and by providing an edge-respecting assignment to all variables in  $V$ . Proposition 2.5 guarantees that  $sv_0 \dots sv_{loop-1} (sv_{loop} \dots sv_k)^\omega$  admits a model  $\sigma$ ; i.e.,  $\sigma \models sv_0 \dots sv_{loop-1} (sv_{loop} \dots sv_k)^\omega$ .

Conversely, if formula  $\phi$  is satisfiable, then automaton  $\mathcal{A}_\phi$  recognizes a nonempty language in  $SV(\phi)^\omega$ . From the Büchi acceptance condition, automaton  $\mathcal{A}_\phi$  recognizes also ultimately periodic locally consistent sequences of the form  $\alpha\beta^\omega$  of length  $k + 1$ , for some finite  $k$  which is bounded by the number of control states of  $\mathcal{A}_\phi$ . Then, by considering the prefix  $\alpha\beta$  we can define an edge-respecting labeling of  $G_{\alpha\beta}$  defining model  $\sigma_k$ .  $\square$

When constraint systems do not have the completion property, locally consistent symbolic models  $\rho$  recognized by automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$  may not admit arithmetical models  $\sigma$  such that  $\sigma \models \rho$ . However, for some constraint systems  $\mathcal{D}$ , it is possible to define a condition  $C$  over symbolic models such that if  $\rho \in \mathcal{L}(\mathcal{A}_s \times \mathcal{A}_\ell)$  satisfies  $C$  then  $\rho$  admits a model. This problem was already studied by [Demri and D'Souza 2007] through an automata-theoretic approach. We show in the next section that it is possible to encode a condition equivalent to  $C$  directly by means of formulae in QF-EUD (when  $\mathcal{D}$  embeds  $\mathbb{N}$  and the successor function).

### 5.1. Checking for $\mathcal{A}_C$

In this section, we provide a direct encoding of a condition of (non) existence of arithmetical models which is equivalent to Property 2.6 on the symbolic model of a CLTLB formula  $\phi$ .

Let  $\lambda$  be the length of symbolic valuations in  $SV(\phi)$  and  $\rho$  be a symbolic model for  $\phi$ . We introduce the notion of point  $p = (x, j, h)$  within  $\rho$  which we use to identify a variable (or a constant)  $x \in V \cup \text{const}(\phi)$  at position  $h$  within symbolic valuation  $\rho(j)$ ; i.e., we refer to variable  $x$ , or constant  $c$ , at position  $j + h$  of the model. Given a point  $p = (x, j, h)$  of  $\rho$ , we denote with  $\text{var}(p)$  the variable  $x$  of  $p$ , with  $sv(p)$  the symbolic valuation  $j$ , and with  $\text{shift}(p)$  the position  $h$  of  $x$  within the  $j$ -th symbolic valuation, also,  $x(j + h)$  is the value of variable  $x$  in position  $h$  of the  $j$ -th symbolic valuation of  $\rho$ .

Given a symbolic model  $\rho$ , we call  $P$  be the set of meaningful points  $p$  of  $\rho$ , i.e., such that  $sv(p) \geq 0$  and  $\text{shift}(p) \in [|\phi|, \lceil \phi \rceil]$ ; when the sequence  $\rho$  of symbolic valuations is of finite length  $k + 1$ , then  $sv(p) \in [0, k + 1]$ , and we indicate the set of points as  $P_k$ .

**Definition 5.12.** We say that there is a *local forward* path between two points  $p_1 = (x, j, h)$  and  $p_2 = (y, i, m)$  of  $\rho$ , written  $p_1 \preceq p_2$ , when  $j = i$ ,  $x(j + h) \leq y(j + m)$  and  $h \leq m$ , for  $x, y \in V$ . A local forward path between  $p_1$  and  $p_2$  is *strict*, written  $p_1 \prec p_2$ , when  $x(j + h) < y(j + m)$ .

Similarly, we say that there is a *local backward* path between two points  $p_1 = (x, j, h)$  and  $p_2 = (y, i, m)$ , written  $p_1 \succeq p_2$ , when  $j = i$ ,  $x(j + h) \leq y(j + m)$  and  $h \geq m$ . A local backward path between  $p_1$  and  $p_2$  is *strict* when  $x(j + h) < y(j + m)$ .

Given an ultimately periodic model  $\alpha\beta^\omega$ , we say that points  $p$  and  $p'$  are *equivalent*, and we write  $p \equiv p'$ , when  $\text{var}(p) = \text{var}(p')$ ,  $sv(p) = sv(p') + k|\beta|$  and  $\text{shift}(p) = \text{shift}(p')$ , for  $k \geq 1$ .

Let  $\rho = \alpha\beta^\omega \in SV(\phi)$  be an ultimately periodic symbolic model of  $\phi$ . The encoding represents, by means of a finite representation, infinite, strict and non strict, paths resulting from iterating infinitely many times suffix  $\beta$ . To do this, we consider the finite path resulting from  $\alpha\beta$ , of length  $k + 1$ , of the form  $\alpha s\beta'$ , with  $\beta = s\beta'$ . Starting from  $\rho(k)$ , we propagate the information about relations  $\prec, \preceq$  among all points representing variables of model  $\rho$ . Forward paths between two points  $p_1, p_2 \in P_k$  are represented by proposition  $F(p_1, p_2)$ , for the strict relation, and  $\tilde{F}(p_1, p_2)$ , for the non strict one. Infinite paths can be represented as “symbolic” cycles originating from relations  $F$  and  $\tilde{F}$  within symbolic valuation  $\rho(\text{loop})$  at the position of loop. The condition for the existence of arithmetic models of  $\rho$  consists in avoiding the existence of a pair of variables  $x, x'$  belonging to the same symbolic valuation at position  $j$ , in suffix  $\beta$ , which are part of an infinite strict (resp. non strict) forward path and an infinite non strict (resp. strict) backward path such that  $x(j + h) < x'(j + m)$ ,

where  $h, m$  are positions of variables within symbolic valuations. Before giving the definition of  $F$  (and  $\tilde{F}$ ), we require a notion of consistency of path which propagates the information of local forward (backward) path from a symbolic valuation at position  $i$  to all adjacent symbolic valuations from position  $i - 1$  to  $i - \lceil \phi \rceil$ . Path consistency between two adjacent symbolic valuations is enforced by the following constraint:

$$F(p_1, p_2) \Leftrightarrow F(p'_1, p_2) \quad (1)$$

for all pairs  $p_1, p_2 \in P_k$ , where  $p_1 = (x, j, h)$ , and  $p'_1$  is

$$p'_1 = \begin{cases} (x, j - 1, h + 1) & h \in [\lfloor \phi \rfloor, \lceil \phi \rceil - 1] \text{ and } j \in [1, k] \\ (x, j + 1, h - 1) & h \in [\lfloor \phi \rfloor + 1, \lceil \phi \rceil] \text{ and } j \in [0, k - 1] \end{cases}$$

where  $p_1$  represents variable  $x$  at position  $h$  within symbolic valuation  $sv(j)$  and  $p'_1$  represents the same variable at position  $h + 1$  (or  $h - 1$ ) within symbolic valuation  $sv(j - 1)$  (or  $sv(j + 1)$ ). Not only  $p_1$  has equivalent points but we need also to consider all the equivalent points to  $p_2$ :

$$F(p_1, p_2) \Leftrightarrow F(p_1, p'_2) \quad (2)$$

for all pair  $p_1, p_2 \in P_k$ , where  $p_2 = (x, j, h)$ , and  $p'_2$  is

$$p'_2 = \begin{cases} (x, j - 1, h + 1) & h \in [\lfloor \phi \rfloor, \lceil \phi \rceil - 1] \text{ and } j \in [1, k] \\ (x, j + 1, h - 1) & h \in [\lfloor \phi \rfloor + 1, \lceil \phi \rceil] \text{ and } j \in [0, k - 1] \end{cases}$$

Consistency rules for  $\tilde{F}$ ,  $B$  and  $\tilde{B}$  are defined similarly.

To verify the existence of a path between two points  $p_1, p_2 \in P_k$ , we check whether there exists a point  $p \in P_k$  which is locally related to  $p_1$  and such that it is connected via a forward path to  $p_2$ . First, observe that only “far” points are considered in the definition of  $F$  or  $\tilde{F}$ . In fact, two points  $p_1$  and  $p_2$  such that  $|sv(p_2) + shift(p_2) - (sv(p_1) + shift(p_1))| \leq -\lfloor \phi \rfloor + \lceil \phi \rceil$  belong to the same symbolic valuation, hence predicate  $F(p_1, p_2)$  (or  $\tilde{F}$ ) can be derived from the local relation  $\preceq$ . By means of rules (1), (2) information is propagated towards symbolic valuations in the past and the future. For instance, let us consider  $\lfloor \phi \rfloor = -1$ ,  $\lceil \phi \rceil = 2$  and a variable  $\{x\}$ . Point  $(x, 3, 2)$  is the same as points  $\{(x, 4, 1), (x, 5, 0), (x, 6, -1)\}$ ; then, they have the same property. While for  $p_1 = (x, 3, 0)$  and  $p_2 = (x, 5, 0)$  predicate  $F(p_1, p_2)$  is  $p_1 \preceq p_2$ , because  $p_2$  is still a point within the symbolic valuation at position 3,  $F(p_1, (x, 4, 2))$  has to be defined by checking whether there is a local point (i.e. within symbolic valuation  $\rho(3)$ ) in relation with  $p_1$  which is, in turn, linked to  $(x, 4, 2)$ . This entails that the definition of  $F$  or  $\tilde{F}$  (and, symmetrically, of  $B$  and  $\tilde{B}$ ) can be given only for “far” points that are such that  $|sv(p_2) + shift(p_2) - (sv(p_1) + shift(p_1))| > -\lfloor \phi \rfloor + \lceil \phi \rceil$ . Predicates  $F$  and  $\tilde{F}$  for all points within the same symbolic valuation are defined by local relation  $\prec$  and  $\preceq$  as follows:

$$\begin{aligned} F(p_1, p_2) &\Leftrightarrow p_1 \prec p_2 \\ \tilde{F}(p_1, p_2) &\Leftrightarrow p_1 \preceq p_2 \end{aligned} \quad (3)$$

where  $sv(p_1) = sv(p_2) \in [0, k]$  and  $shift(p_1), shift(p_2) \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ . Predicates  $F$  and  $\tilde{F}$  for far points are:

$$\begin{aligned} F(p_1, p_2) &\Leftrightarrow \bigvee_{p \in P_k, p \neq p_1} \left( (p_1 \prec p \wedge \tilde{F}(p, p_2)) \vee (p_1 \preceq p \wedge F(p, p_2)) \right) \\ \tilde{F}(p_1, p_2) &\Leftrightarrow \bigvee_{p \in P_k, p \neq p_1} p_1 \preceq p \wedge \tilde{F}(p, p_2) \end{aligned} \quad (4)$$

for all  $p_1, p_2$  such that  $sv(p_2) = i \in [1, k]$ ,  $sv(p_1) \in [0, i - 1]$ ,  $|sv(p_2) + shift(p_2) - (sv(p_1) + shift(p_1))| > -\lfloor \phi \rfloor + \lceil \phi \rceil$ , and  $shift(p_1) = \lfloor \phi \rfloor$ . Notice that considering  $shift(p_1) = \lfloor \phi \rfloor$

suffices to define correctly predicates  $F$  and  $\tilde{F}$ . In fact, with reference to Figure 1, all points within a symbolic valuation  $\rho(i)$  (with  $i \in [0, k]$ ) are covered by the local relation (3) (e.g., points  $(y, i, -1)$  and  $(y, i, 0)$ , where the local relation is represented through a solid line) while for any  $x$ , all positions from  $(x, 0, \lfloor \phi \rfloor)$  to  $(x, k-1, \lfloor \phi \rfloor)$  are covered by (4) through argument  $p_1$  when  $sv(p_1) \in [0, k-1]$  and  $shift(p_1) = \lfloor \phi \rfloor$  (e.g.  $p_1$  and  $p_2$  in Figure 1).

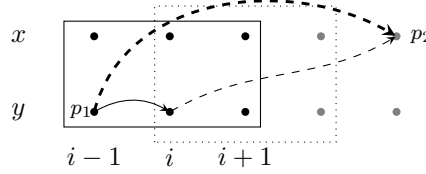


Fig. 1. Adjacent symbolic valuations  $\rho(i)$  (solid line) and  $\rho(i+1)$  (dotted line) of length 2 not covering both far points  $p_1 = (y, i, -1)$  and  $p_2 = (x, j, h)$  (with  $j > i$  and  $-1 \leq h \leq 1$ ) of the model.

In Figure 1 we show how propositions  $F/\tilde{F}$  involving points within symbolic valuations (rectangles) are defined by means of  $\prec/\preceq$  (continuous line) whereas “far” points require the conjunction of a local relation (continuous line) and a relation  $F/\tilde{F}$  (thin dotted line). Moreover, when points  $p_1, p_2$  can not be linked by a forward/backward path, i.e. when  $sv(p_1) + shift(p_1) > sv(p_2) + shift(p_2)$ , then:

$$F(p_1, p_2) \Leftrightarrow \perp \text{ and } \tilde{F}(p_1, p_2) \Leftrightarrow \perp. \quad (5)$$

We can now define a condition for the existence of infinite paths resulting from iterating infinitely many times suffix  $v$  of the ultimately periodic model  $\alpha\beta^\omega$  of  $\phi$ . Let  $loop \in [1, k]$  be the position of the loop, i.e., the position of symbolic valuation  $s$  in the finite word  $\alpha s\beta'$  representing  $\rho$ . An infinite forward (resp. backward) path can be represented as a cycle among variables belonging to a symbolic valuation  $\rho(loop-1) = \rho(k)$ , by means of predicates  $F$  and  $\tilde{F}$ .

We define  $LF(p)$  (resp.  $\tilde{LF}(p)$ ) as an abbreviation for the condition of existence of a strict (resp. non-strict) cycle (loop forward) on  $p$  in symbolic valuation  $\rho(loop-1)$  (notice that for each  $p$  s.t.  $sv(p) = loop-1$  there is only one  $p' \in P_k$  s.t.  $p \equiv p'$ , and it is  $sv(p') = k$ ):

$$\begin{aligned} LF(p) &:= sv(p) = loop - 1 \wedge \forall p' \in P_k (p \equiv p' \Rightarrow F(p, p')) \\ \tilde{LF}(p) &:= sv(p) = loop - 1 \wedge \forall p' \in P_k (p \equiv p' \Rightarrow \tilde{F}(p, p')). \end{aligned}$$

**PROPOSITION 5.13.** *Let  $\rho = \alpha\beta^\omega \in SV(\phi)^\omega$  be an ultimately periodic word, there exists a non-strict (resp. strict) infinite forward path in  $\rho$  involving point  $p$ , with  $var(p) \in V$ ,  $sv(p) = loop-1$ , and  $shift(p) \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ , if, and only if,  $\tilde{LF}(p)$  (resp.  $LF(p)$ ).*

**PROOF.** Let us suppose that there exists an infinite (non strict) forward path  $w$  in  $\rho$  and the suffix  $\beta$  is of the form  $s\beta'$ . We can consider that  $w$  starts from  $s$  without loss of generality, as  $\alpha$  is a finite prefix. Let  $n = |V| \cdot (-\lfloor \phi \rfloor + \lceil \phi \rceil + 1)$  be the number of points within symbolic valuations. Let us consider the word  $\beta^{n^2}$ , where the suffix  $\beta$  is repeated  $n^2$  times. Let  $p_i$  be a point such that  $sv(p) = i$  and  $u$  be the position of an occurrence of  $s$  in  $\rho$  and  $\preceq^*$  be the transitive closure of  $\preceq$ . We represent the sequence of points which are visited by  $w$  at each occurrence of  $s$  in  $\rho$  as:

$$p_u \preceq^* p_{u+|\beta|} \preceq^* \cdots \preceq^* p_{u+l|\beta|}$$

where all points  $p_{u+l|\beta|}$ , with  $l \in [0, n^2]$ , are in  $w$ . Therefore, since the number  $n$  of points within symbolic valuations at each position of  $\rho$  is finite, then there exists a position  $j \leq n^2$  such that  $p_u \equiv p_{u+j|\beta|}$ ; i.e.,  $w$ , passing through  $\beta^{n^2}s$ , visits eventually point  $p_{u+j|\beta|}$  which is equivalent

to  $p_u$ . Hence, by transitivity  $p_u \preceq^* p_{u+j|\beta|}$ . Moreover, let us consider all points  $q_{u+l|\beta|}$ , with  $l \in [0, n^2]$ , such that  $p_u \equiv q_{u+l|\beta|}$ . Then, because suffix  $\beta$  is iterated and so are all the relations between two points belonging to each  $\beta s$  which occurs in  $\rho$ , and because  $p_u \preceq^* p_{u+j|\beta|}$  we have that  $q_{u+l|\beta|} \preceq^* q_{u+(l+1)|\beta|}$  for all  $l \in [0, n^2 - 1]$  where each relation  $\preceq^*$  between  $q_{u+l|\beta|}$  and  $q_{u+(l+1)|\beta|}$  is witnessed by a path within the corresponding occurrence of  $\beta s$  which is recursively defined as  $q_{u+l|\beta|} \preceq \bar{p}_{u+l|\beta|} \preceq^* q_{u+(l+1)|\beta|}$ , for some  $\bar{p}_{u+l|\beta|}$ . By definition of  $\tilde{F}$  over the prefix  $\rho' = \alpha s \beta' s$  of  $\rho$  we have  $\tilde{F}(p, p') = p \preceq p'' \wedge \tilde{F}(p'', p')$  where points  $p, p' \equiv p_u$  are representative of all points  $q_{u+l|\beta|}$ ,  $sv(p) = loop - 1$  and  $p'(loop - 1) = s$ ,  $sv(p') = k$  and  $\rho'(k) = s$  and  $p'' \equiv \bar{p}_{u+l|\beta|}$ .

Conversely, when  $F(p, p')$  holds, with  $p = (x, loop - 1, h)$  and  $p' = (x, k, h)$  by definition, we have:

$$\tilde{F}(p, p') \Leftrightarrow p \preceq p'' \wedge \tilde{F}(p'', p').$$

Recursively, each term  $\tilde{F}(p'', p')$  entails a path from  $p''$  to  $p'$  such that  $p'' \preceq^* p'$ . The infinite path visits infinitely often all points of the path  $p \rightarrow p'' \rightarrow \dots \rightarrow p'$ . Since  $p \equiv p'$  and the suffix  $\beta$  is repeated infinitely often, then the sequence of points  $p \rightarrow p'' \rightarrow \dots \rightarrow p'$  is visited is infinitely many times. Therefore, point  $p$  belongs to an infinite forward path along  $\rho = \alpha \beta^\omega$ .

In case of infinite strict forward paths, the previous arguments can be adapted as follows. The first part of the proof has to be modified in the length of the word  $\beta^t$  which one has to consider to find the occurrence of a point in  $w$  which is equivalent to  $p_u$ . By taking  $t$  big enough (at most the number of all possible paths between two points in  $\beta$  for all  $n^2$  pairs of points), it is possible to find a suitable position  $j \leq t$  such that  $p_{u+j|\beta|} \equiv p_u$ . The second side of the implication is proved by considering the two cases  $F(p, p') \Leftrightarrow p \prec p'' \wedge \tilde{F}(p'', p')$  and  $F(p, p') \Leftrightarrow p \preceq p'' \wedge F(p'', p')$ .  $\square$

Analogously, we can define predicates  $B, \tilde{B}$  for backward relations,  $LB$  and  $\tilde{LB}$  for backward cycles. Proposition 5.13 can be given also in case of backward paths.

**PROPOSITION 5.14.** *Let  $\rho = \alpha \beta^\omega \in SV(\phi)^\omega$  be an ultimately periodic word. There exists a non-strict (resp. strict) infinite backward path in  $\rho$  involving point  $p$ , with  $var(p) \in V$ ,  $sv(p) = loop - 1$ , and  $shift(p) \in [\llbracket \phi \rrbracket, \lceil \phi \rceil]$ , if and only if,  $\tilde{LB}(p)$  (resp.  $LB(p)$ ).*

Our condition for the non existence of an arithmetic model follows immediately from Definition 2.6 and by previous Propositions 5.13 and 5.14. The condition holds when there exists a symbolic valuation  $\rho(j)$ , with  $j \in [loop - 1, k]$ , such that a strict (resp. non strict) forward path and a non strict (resp. strict) backward path are linked together by means of a strict edge  $<$ .

$$\exists p_1, p_2, p'_1, p'_2, \bar{p}_f, \bar{p}_b \left( \begin{array}{c} \tilde{F}(p_1, \bar{p}_f) \wedge \tilde{F}(\bar{p}_f, p_2) \wedge T^B(p'_1, \bar{p}_b, p'_2) \wedge (\bar{p}_f \prec \bar{p}_b \vee \bar{p}_f \succ \bar{p}_b) \\ \vee \\ \tilde{B}(p'_1, \bar{p}_b) \wedge \tilde{B}(\bar{p}_b, p'_2) \wedge T^F(p_1, \bar{p}_f, p_2) \wedge (\bar{p}_f \prec \bar{p}_b \vee \bar{p}_f \succ \bar{p}_b) \end{array} \right) \quad (6)$$

where  $p_1, p_2, p'_1, p'_2, \bar{p}_f, \bar{p}_b \in P_k$  such that  $p_1 \equiv p_2$ ,  $p'_1 \equiv p'_2$ ,  $p_1 \neq p'_1$  and  $\bar{p}_f, \bar{p}_b$  are such that  $sv(\bar{p}_f), sv(\bar{p}_b) \in [loop - 1, k]$ .

Predicates  $T^F(p_1, \bar{p}, p_2)$  and  $T^B(p_1, \bar{p}, p_2)$  formalize the existence of, respectively, strict forward and strict backward paths between  $p_1$  and  $p_2$  visiting  $\bar{p}$ . They are defined as follows, for all points  $p_1, p_2, \bar{p} \in P_k$  such that  $sv(p_1) = loop - 1$ ,  $sv(p_2) = k$  and  $sv(\bar{p}) \in [loop, k]$ :

$$\begin{aligned} T^B(p_1, \bar{p}, p_2) &\Leftrightarrow \left( \left( \tilde{B}(p_1, \bar{p}) \wedge B(\bar{p}, p_2) \right) \vee \left( B(p_1, \bar{p}) \wedge \tilde{B}(\bar{p}, p_2) \right) \right) \\ T^F(p_1, \bar{p}, p_2) &\Leftrightarrow \left( \left( \tilde{F}(p_1, \bar{p}) \wedge F(\bar{p}, p_2) \right) \vee \left( F(p_1, \bar{p}) \wedge \tilde{F}(\bar{p}, p_2) \right) \right). \end{aligned}$$

Notice that  $\tilde{F}(p_1, \bar{p}_f) \wedge \tilde{F}(\bar{p}_f, p_2)$  in (6) implies  $\tilde{LF}(p_1)$ , while  $T^F(p_1, \bar{p}_f, p_2)$  implies  $LF(p_1)$ . Similarly for  $\tilde{LB}(p'_1)$  and  $LB(p'_1)$ .

*Remark 5.15.* Though different from Property 2.6, the condition of nonexistence of arithmetic models formalized by formula (6) is equivalent to the former, as discussed in the following.

We summarize condition (6) to make the comparison easier:

- there is an infinite forward path  $f$  from  $p_1$ , where  $sv(p_1) = loop - 1$  (witnessed by either  $\tilde{LF}(p_1)$  or  $LF(p_1)$  implied in (6));
- there is an infinite backward path  $b$  from  $p_2$ , where  $sv(p_2) = loop - 1$  (witnessed by either  $\tilde{LB}(p'_1)$  or  $LB(p'_1)$  implied in (6));
- either  $f$  or  $b$  are strict;
- there are two points  $\bar{p}_f$  of  $f$  and  $\bar{p}_b$  of  $b$  such that  $sv(\bar{p}_f) = sv(\bar{p}_b)$  and  $\bar{p}_f < \bar{p}_b$  or  $\bar{p}_f > \bar{p}_b$ .

In particular, Part 4 of Property 2.6 is slightly different, since it states that for each  $i, j \in \mathbb{N}$ , given a forward path  $d$  and a backward path  $e$ , whenever  $d(i)$  and  $e(j)$  belong to the same symbolic valuation there is an edge labeled by  $<$  from  $d(i)$  to  $e(j)$ . In other words, this means that point  $p_d$  representing  $d(i)$  and point  $p_e$  representing  $e(j)$  are such that either  $p_d < p_e$  (if  $sv(p_d) + shift(p_d) \leq sv(p_e) + shift(p_e)$ ) or  $p_d > p_e$  (if  $sv(p_d) + shift(p_d) \geq sv(p_e) + shift(p_e)$ ). Observe that Property 2.6 is defined for a general  $G_\rho$  while our condition (6) is adapted to the finite representation of ultimately periodic symbolic models  $\rho = \alpha\beta^\omega$ .

Let us consider that condition (6) holds. Therefore, there exists a pair of points  $p_1$  and  $p'_1$ , such that  $sv(p_1) = sv(p'_1) = loop - 1$ , visited respectively by an infinite forward path, including point  $\bar{p}_f$ , and an infinite backward path, including point  $\bar{p}_b$ , such that  $\bar{p}_f < \bar{p}_b$  or  $\bar{p}_f > \bar{p}_b$ . By transitivity, this immediately entails  $p_1 < p'_1$  or  $p_1 > p'_1$ . Now, we have to consider two cases. Let us consider any two points  $u$  and  $v$  such that  $sv(u) = sv(v) \leq sv(p_1)$  (we consider two points in the prefix  $\alpha sv_{loop}$ ). If  $u$  is connected to  $p_1$  by a forward path, i.e.,  $F(u, p_1)$  or  $\tilde{F}(u, p_1)$ , and  $v$  is connected to  $p_2$  by a backward path, i.e.,  $B(v, p_2)$  or  $\tilde{B}(v, p_2)$ , then  $u < v$  or  $u > v$  (and so we obtain condition 4 of Property 2.6). In the second case, we choose two points  $u, v$  belonging to the same symbolic valuation in the suffix  $\beta$  which are visited by a forward and a backward path, respectively, i.e.,  $F(u, p_2)$  or  $\tilde{F}(u, p_2)$  and  $B(v, p'_2)$  or  $\tilde{B}(v, p'_2)$ , where  $p_1 \equiv p_2$  and  $p'_1 \equiv p'_2$ . Again, since  $\beta$  is repeated infinitely many times then it must be  $u < v$  or  $u > v$ , because there exist a forward path from  $u$  to  $p_2$  and from  $p_2$  to a point  $\hat{p}_f \equiv \bar{p}_f$  and a backward path from  $v$  to  $p'_2$  and from  $p'_2$  to a point  $\hat{p}_b \equiv \bar{p}_b$  such that  $\hat{p}_f < \hat{p}_b$  or  $\hat{p}_f > \hat{p}_b$ .

Conversely, if Property 2.6 holds, then there exist a forward path and a backward path which have two points  $\bar{p}_f$  and  $\bar{p}_b$  such that  $\bar{p}_f < \bar{p}_b$ , as shown in the proof of Lemma 6.2 of [Demri and D'Souza 2007]. Essentially, this is a consequence of the fact that if  $\rho$  does not admit a model, then there are two points  $u, v$  which can be connected together by a path which contains an infinite number of strict relations  $<$ . Since  $\rho$  is ultimately periodic, and the number of pairs of points such that  $\bar{p}_f \not\equiv \bar{p}_b$  is finite, by choosing an appropriate number of iterations of  $\beta$  there must be two equivalent points which are connected with by a strict path. This is witnessed by our condition (6), in particular, by looking for two points  $\bar{p}_f, \bar{p}_b$  belonging to a forward path and a backward path which are connected, i.e.,  $\bar{p}_f < \bar{p}_b$  or  $\bar{p}_f > \bar{p}_b$ .

We have the next theorem, which extends Proposition 5.11 to constraint system  $IPC^*$ , which does not benefit of the completion property.

**THEOREM 5.16.** *Let  $\phi \in CLTLB(\mathcal{D})$  and  $\mathcal{D}$  be  $IPC^*$ . Then,  $\phi$  is  $k$ -satisfiable and formula (6) does not hold if, and only if, formula  $\phi$  has a model  $\alpha\beta^\omega$ , with  $|\alpha\beta| = k + 1$ .*

**PROOF.** By Theorem 5.10,  $\phi$  is  $k$ -satisfiable if, and only if, formula  $|\phi|_k$  is satisfiable. By hypothesis of the theorem, formula  $|\phi|_k$  induces a model  $\sigma_k$ . Formula (6) constrains values of model  $\sigma_k$ , being a set of formulae over values of variables defined by sequence  $\alpha\beta$  of symbolic valuations



of length  $k$ . By Theorem 5.8 symbolic model  $\rho$  is such that  $\rho \models^{sym} \phi$ . Finally, since formula (6) does not hold then sequence  $\rho$  admits a model  $\sigma$  such that  $\sigma \models \phi$ , by Proposition 2.7. Model  $\sigma$  can be obtained from  $\sigma_k$  by iterating suffix  $v$  and by providing an edge-respecting labeling of  $G_\rho$ .

Conversely, if formula  $\phi$  is satisfiable, then automaton  $\mathcal{A}_\phi$  recognizes models which satisfy condition  $C$ . Then,  $k$ -models can be obtained as in the proof of Proposition 5.11.  $\square$

In order to encode the previous formulae into QF-EUD formulae, where  $\mathcal{D}$  is a suitable constraint system embedding  $\mathbb{N}$  and having the successor function plus order  $<$ , we rearrange the formulae above by splitting information, which is now encapsulated in the notion of point, on variables and positions over the model. To encode local forward paths we introduce predicate  $f_{x,y} : \mathbb{N}^3 \rightarrow \{true, false\}$  for all pairs  $x, y \in V \cup const(\phi)$  (and similarly  $\tilde{f}$ ) to encode relation  $p_1 \prec p_2$  ( $p_1 \preceq p_2$ ) where  $p_1 = (x, j, h)$  and  $p_2 = (y, j, m)$ .

$f_{x,y}$	$0 \leq j \leq k$ and $h \leq m$	$0 \leq j \leq k$ and $h > m$
$f_{x,y}(j, h, m)$	$\mathbf{f}_{x,y} \Leftrightarrow \sigma_k(j + h, x) < \sigma_k(j + m, y)$	$\mathbf{f}_{x,y} \Leftrightarrow \perp$
$\tilde{f}_{x,y}(j, h, m)$	$\tilde{\mathbf{f}}_{x,y} \Leftrightarrow \sigma_k(j + h, x) \leq \sigma_k(j + m, y)$	$\tilde{\mathbf{f}}_{x,y} \Leftrightarrow \perp$

for all  $h, m \in [[\phi], \lceil \phi \rceil]$ . When both  $x, y \in const(\phi)$  then  $f_{x,y} \Leftrightarrow x < y$  and  $\tilde{f}_{x,y} \Leftrightarrow x \leq y$  for all  $0 \leq j \leq k$  and  $h \leq m$ ;  $f_{x,y} \Leftrightarrow \perp$  and  $\tilde{f}_{x,y} \Leftrightarrow \perp$  for all  $0 \leq j \leq k$  and  $h > m$ . Notice that the value of  $\sigma_k(0 + h, x)$  equals the value of term  $\alpha = Y^{|h|}x$ , for  $h \in [[\phi], -1]$ , or of term  $\alpha = X^h x$ , for  $h \in [0, \lceil \phi \rceil]$ . Then, the value of  $\sigma_k(0 + h, x)$  is  $\alpha(0)$ , and similarly  $\sigma_k(k + h, x)$  is  $\alpha(k)$  (see  $|ArithConstraints|_k$  in Section 4). Observe that constants are implicitly included in the model. For instance, if  $5 \in const(\phi)$  and  $x \in V$  we have the following formulae  $f_{x,5}(j, h, m) \Leftrightarrow \sigma_k(j + h, x) < 5$  and  $\tilde{f}_{5,x}(j, h, m) \Leftrightarrow 5 < \sigma_k(j + m, x)$ .

Predicates  $F$  are encoded by uninterpreted predicates  $F_{x,y} : \mathbb{N}^4 \rightarrow \{true, false\}$  for all pairs of variables  $x, y \in V \cup const(\phi)$ . Consistency of predicate  $F$  is then enforced by formula  $|ConsistencyConstraints|_k$ :

$i \in [1, k]$	$m \in [[\phi], \lceil \phi \rceil]$	$j$
$\mathbf{F}_{x,y}(j, h, i, m) \Leftrightarrow \mathbf{F}_{x,y}(j + 1, h - 1, i, m)$	$h \in [[\phi] + 1, \lceil \phi \rceil]$	$[0, i - 1]$
$\mathbf{F}_{x,y}(j, h, i, m) \Leftrightarrow \mathbf{F}_{x,y}(j - 1, h + 1, i, m)$	$h \in [[\phi], \lceil \phi \rceil - 1]$	$[1, i]$

and

$j \in [0, k - 1]$	$h \in [[\phi], \lceil \phi \rceil]$	$i$
$\mathbf{F}_{x,y}(j, h, i, m) \Leftrightarrow \mathbf{F}_{x,y}(j, h, i + 1, m - 1)$	$m \in [[\phi] + 1, \lceil \phi \rceil]$	$i \in [j, k - 1]$
$\mathbf{F}_{x,y}(j, h, i, m) \Leftrightarrow \mathbf{F}_{x,y}(j, h, i - 1, m + 1)$	$m \in [[\phi], \lceil \phi \rceil - 1]$	$i \in [j + 1, k]$

Formulae defining  $F$  are encoded as follows:

$$\mathbf{F}_{x,y}(j, h, i, m) \Leftrightarrow \begin{cases} \bigvee_{z \in V} \bigvee_{u = \lfloor \phi \rfloor}^{\lceil \phi \rceil} \mathbf{f}_{x,z}(j, h, u) \wedge \tilde{\mathbf{F}}_{z,y}(j, u, i, m) \vee \\ \bigvee_{z \in V} \bigvee_{u = \lfloor \phi \rfloor}^{\lceil \phi \rceil} \tilde{\mathbf{f}}_{x,z}(j, h, u) \wedge \mathbf{F}_{z,y}(j, u, i, m) \end{cases}$$

$$\tilde{\mathbf{F}}_{x,y}(j, h, i, m) \Leftrightarrow \bigvee_{z \in V} \bigvee_{u = \lfloor \phi \rfloor}^{\lceil \phi \rceil} \tilde{\mathbf{f}}_{x,z}(j, h, u) \wedge \tilde{\mathbf{F}}_{z,y}(j, u, i, m)$$

for all  $j, i \in [0, k]$  with  $j < i$  and for all  $h, m \in [[\phi], \lceil \phi \rceil]$  such that  $j + h \leq i + m$ ,  $i + m - (j + h) > -\lfloor \phi \rfloor + \lceil \phi \rceil$ ,  $h = \lfloor \phi \rfloor$ ,  $(x = z) \Rightarrow (h \neq u)$  and for all pair  $x, y \in V \cup const(\phi)$ . When  $j = i \in [0, k]$

and  $h \leq m$ , with  $h, m \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ :

$$\mathbf{F}_{x,y}(j, h, j, m) \Leftrightarrow \mathbf{f}_{x,y}(j, h, m)$$

$$\tilde{\mathbf{F}}_{x,y}(j, h, j, m) \Leftrightarrow \tilde{\mathbf{f}}_{x,y}(j, h, m)$$

When  $j + h > i + m$  then:

$$\mathbf{F}_{x,y}(j, h, i, m) \Leftrightarrow \perp$$

$$\tilde{\mathbf{F}}_{x,y}(j, h, i, m) \Leftrightarrow \perp$$

Figure 2 how predicate  $F_{x,x}(i, 1, j, 1)$  is defined as conjunction of local relation  $f_{x,y}(i, 0, 1)$  and  $F_{y,x}(i, 1, j, 1)$ .

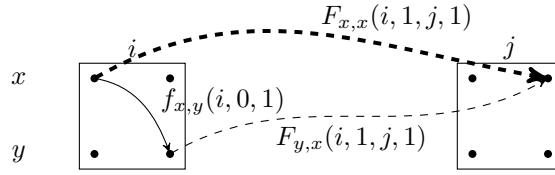


Fig. 2. Definition of  $F$  by local relations  $f$ .

Predicates capturing the definitions of backward loops are defined similarly (see Appendix 9.4 for further details).

We introduce the following abbreviations, which capture infinite forward (backward) paths that form a cycle at positions  $loop - 1$  and  $k$ .

$$\mathbf{LF}_x(h) := \mathbf{F}_{x,x}(loop - 1, h, k, h)$$

$$\tilde{\mathbf{LF}}_x(h) := \tilde{\mathbf{F}}_{x,x}(loop - 1, h, k, h)$$

for all  $x \in V$  and  $h \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ . By definition of backward paths, abbreviations  $LB$  and  $\tilde{LB}$  are instead defined between positions  $k$  back to  $loop - 1$ :

$$\mathbf{LB}_x(h) := \mathbf{B}_{x,x}(k, h, loop - 1, h)$$

$$\tilde{\mathbf{LB}}_x(h) := \tilde{\mathbf{B}}_{x,x}(k, h, loop - 1, h)$$

Finally, our condition is encoded by the following QF-EUD formula. The condition is expressed with reference to a pair of elements  $x, x' \in V \cup \text{const}(\phi)$ , hence it is parametric with respect to them. The condition is meaningful only if  $x \neq x'$  and if either  $x \notin \text{const}(\phi)$  or  $x' \notin \text{const}(\phi)$ . In fact, a constant value never generates a strict (forward or backward) path; therefore, two constants can not satisfy the condition of non-existence of an arithmetical model. Also, for  $p = (x, loop - 1, h)$  it cannot be  $\tilde{LF}(p) \wedge LB(p)$  nor  $LF(p) \wedge \tilde{LB}(p)$ . The formula  $C_{x,x'}(i)$  below captures the existence in  $\rho(i)$  of a strict relation  $<$  between two points, one of a forward and one of backward path, which involve variables  $x$  and  $x'$ :

$$C_{x,x'}(i) := \bigvee_{y, y' \in V \cup \text{const}(\phi)} \exists h, h', n, n' \left( \begin{array}{l} \tilde{\mathbf{F}}_{x,y}(loop - 1, h, i, n) \wedge \tilde{\mathbf{F}}_{y,x}(i, n, k, h) \wedge \\ \mathbf{T}_{x',y'}^B(h', i, n') \wedge \\ (\mathbf{f}_{y,y'}(i, n, n') \vee \mathbf{b}_{y,y'}(i, n, n')) \\ \vee \\ \tilde{\mathbf{B}}_{x,y}(i, n, loop - 1, h) \wedge \tilde{\mathbf{B}}_{y,x}(k, h, i, n) \wedge \\ \mathbf{T}_{x',y'}^F(h', i, n') \wedge \\ (\mathbf{f}_{y',y}(i, n', n) \vee \mathbf{b}_{y',y}(i, n', n)) \end{array} \right)$$

where  $h, h', n, n' \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ . Predicates  $T^F$  and  $T^B$  are essentially shorthands defined as follows:

$$\begin{aligned} T_{x,y}^B(h, i, n) &\Leftrightarrow \begin{cases} \tilde{B}_{x,y}(k, h, i, n) \wedge B_{y,x}(i, n, \text{loop} - 1, h) \\ \vee \\ B_{x,y}(k, h, i, n) \wedge \tilde{B}_{y,x}(i, n, \text{loop} - 1, h) \end{cases} \\ T_{x,y}^F(h, i, n) &\Leftrightarrow \begin{cases} \tilde{F}_{x,y}(\text{loop} - 1, h, i, n) \wedge F_{y,x}(i, n, k, h) \\ \vee \\ F_{x,y}(\text{loop} - 1, h, i, n) \wedge \tilde{F}_{y,x}(i, n, k, h). \end{cases} \end{aligned}$$

The existence condition of an arithmetical model is captured by the formula:

$$\bigwedge_{\substack{x, x' \in V \cup \text{const}(\phi) \\ x \neq x', x \notin \text{const}(\phi) \vee x' \notin \text{const}(\phi)}} \forall i (\text{loop} - 1 \leq i \leq k \Rightarrow \neg C_{x,x'}(i)) \quad (7)$$

The universal quantifier ranges over a finite domain: the set of time positions from  $\text{loop} - 1$  and  $k$ . Therefore, formula 7 is actually a QF-EU( $\mathbb{N}, <$ ) formula because quantifiers can be represented by conjunctions, since they range over finite sets of elements. However, we can devise a simpler condition than the previous one, which avoids using the universal quantifier over time positions  $i$ . In fact, as mentioned in Remark 5.15, it can be shown that, given a pair  $x, x'$ , there is a position  $i \in [\text{loop} - 1, k]$  such that  $C_{x,x'}(i)$  if and only if *for all*  $i \in [\text{loop} - 1, k]$  it is  $C_{x,x'}(i)$ . In other words, we have that for all  $i \in [\text{loop} - 1, k]$  it is  $\neg C_{x,x'}(i)$  if and only if *there is at least one*  $i \in [\text{loop} - 1, k]$  such that  $\neg C_{x,x'}(i)$ . Then, for each pair  $x, x'$  we introduce an arithmetic term  $i_{xx'}$ , and we build the following QF-EU( $\mathbb{N}, <$ ) constraint:

$$\bigwedge_{\substack{x, x' \in V \cup \text{const}(\phi) \\ x \neq x', x \notin \text{const}(\phi) \vee x' \notin \text{const}(\phi)}} (\text{loop} - 1 \leq i_{xx'} \leq k) \wedge \neg C_{x,x'}(i_{xx'}). \quad (8)$$

Both formulae (7) and (8) are quantifier-free, but formula (8) avoids also the explicit finite quantification over position  $i$ ; also, it allows to exploit the features of satisfiability solvers: when (8) is solved, for each pair  $x, x'$ , the solver looks for a value to assign to  $i_{xx'}$ ; if it cannot find any, then the formula is unsatisfiable, and no arithmetic models exist.

Finally, given a CLTLB(IPC\*) formula  $\phi$ , we feed the solver the following QF-EU(IPC\*) formula:

$$|\phi|_k \wedge (8). \quad (9)$$

If QF-EUF formula (9) is unsatisfiable, then either  $\phi$  does not admit symbolic models, or none of its symbolic models admit arithmetic models. Conversely, if QF-EUF formula (9) is satisfiable, then there is a symbolic model  $\rho$  of  $\phi$  for which condition (8) holds, hence  $\rho$  admits an arithmetic model and  $\phi$  is satisfiable.

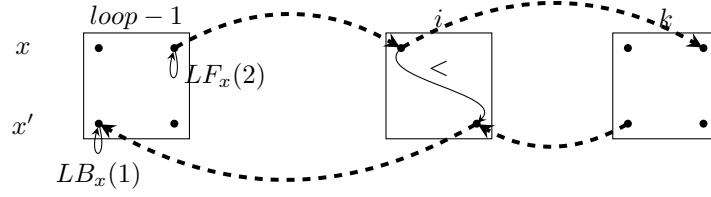
Figure 3 shows an example of a model satisfying our non existence condition  $C_{x,x'}(i)$ .

## 5.2. Complexity

In this section we provide an estimation of the size of the formulae constituting the encoding of Section 4, including, where they are needed, the constraints of Section 5.1.

The encoding of Section 4 is linear in the size of the formula  $\phi$  (and of the bound  $k$ ). In fact, if  $m$  is the total number of subformulae and  $n$  is the total number of temporal operators **U** and **R** occurring in  $\phi$ , the QF-EUD encoding requires  $n + 1$  integer variables (one each for  $\text{loop}$  and the  $j_\psi$ 's) and  $m$  unary predicates (one for each subformula in  $cl(\phi)$ ).

The total size of the formulae in Section 5.1 is polynomial in bound  $k$ , in the cardinality of the set of variables and constants, and in the size of symbolic valuations. In fact, the encoding of the

Fig. 3. Model satisfying the non existence condition  $C_{x,x'}(i)$ .

condition for the existence of an arithmetical model requires a QF-EU( $\mathbb{N}, <, =$ ) formula of size quadratic in the length  $k$  and in the number  $|V|$  of variables, but double quadratic in the size of symbolic valuations.

Let  $\lambda$  be the size  $\lambda = \lceil \phi \rceil + \lfloor \phi \rfloor + 1$  of symbolic valuations and  $V'$  be the set  $V \cup \text{const}(\phi)$ . The total number of non-trivial predicates  $f_{x,y}, \tilde{f}_{x,y}$  (resp.  $f_{x,y}, \tilde{f}_{x,y}$ ), i.e., those where  $h \leq m$ , is defined by the following parametric formula:

$$N(a, b) = (k+1) \sum_{i=1}^{\lambda} |a| \cdot ((\lambda - i) + (|b| - 1) \cdot (\lambda - i + 1)) = \\ (k+1) \left( |a||b| \frac{\lambda(\lambda+1)}{2} - |a|\lambda \right).$$

Each predicate has fixed dimension and the number of non-trivial ones results from the sum of the following three cases:

- $x, y \in V$ , which is  $N(|V|, |V|)$
- $x \in V, y \in \text{const}(\phi)$ , which is  $N(|V|, |\text{const}(\phi)|)$
- $x \in \text{const}(\phi), y \in V$ , which is  $N(|\text{const}(\phi)|, |V|)$ .

that is bounded by  $N_{\text{local}} = N(|V'|, |V'|) \leq (k+1)|V'|^2\lambda^2$ .

In order to compute the size of formulae  $F, \tilde{F}$  (resp.  $B, \tilde{B}$ ) we first determine the number of pairs of points for which  $F_{x,y}(j, h, i, m)$  is not trivially false. The following function  $N_{p,p'}$

$$N_{p,p'} = |V'| \sum_{i=\lfloor \phi \rfloor}^{k+\lceil \phi \rceil} |V'| (k + \lceil \phi \rceil - i) = |V'|^2 \sum_{i=0}^{k+\lambda-1} i = |V'|^2 \frac{(k+\lambda-1)(k+\lambda)}{2} \leq |V'|^2 (k+\lambda)^2$$

corresponds to the number of pairs of points  $p, p'$  that generate non-trivial predicates  $F, \tilde{F}$  (resp.  $B, \tilde{B}$ ) because their position is such that  $sv(p_1) + \text{shift}(p_1) \leq sv(p_2) + \text{shift}(p_2)$  (resp.  $sv(p_1) + \text{shift}(p_1) \geq sv(p_2) + \text{shift}(p_2)$ ). We compute the size of (non-trivial) formulae  $F, \tilde{F}$  (and  $B, \tilde{B}$ ) by counting the number of subformulae involved in their definition. We consider only the case for  $F$  because the others have the same (worst) complexity. Each formula  $F$  involves, in the worst case (i.e., for points that do not belong to the same symbolic valuation),  $|V| - 1$  variables  $z \in V$  with respect to  $\lambda$  different positions  $u$ . Then, an instance of  $F$  requires at most  $(|V| - 1)\lambda$  disjuncts. The upper bound for the total size of all formulae defining predicates  $F, \tilde{F}$  (resp.  $B, \tilde{B}$ ) is

$$N_{far} = N_{p,p'} 2(|V| - 1)\lambda \leq \lambda |V| |V'|^2 (k + \lambda)^2 \leq \lambda |V'|^3 (k + \lambda)^2.$$

The analysis of formulae  $|\text{ConsistencyConstraints}|_k$  shows that each point belongs to  $\lambda$  symbolic valuations (e.g., if  $\lceil \phi \rceil = 0, \lfloor \phi \rfloor = 1$ , then  $\lambda = 2$ , and points  $(x, 4, 1)$  and  $(x, 5, 0)$  correspond to the same element), and for all pairs  $p_1, p_2$  we define consistency among the  $\lambda$  points correspond-

ing to  $p_1$  and the  $\lambda$  points corresponding to  $p_2$ . Therefore, we need at most

$$N_{CC} = \lambda^2 |V'|^2 \frac{(k + \lambda - 1)(k + \lambda)}{2} \leq \lambda^2 |V'|^2 (k + \lambda)^2$$

constraints  $|ConsistencyConstraints|_k$ , where each formula is of fixed dimension.

Finally, predicate  $C_{x,x'}(i)$  appears in formula (8) once for each  $|V'| |V| \lambda^2$  pairs of points  $x, x'$ . In addition, each instance of  $C_{x,x'}(i)$  has  $(|V'| \lambda)^2$  disjuncts, one for each possible  $\bar{p}_f, \bar{p}_b$  of formula 6. Index  $i$  instead is a free variable in (8) and we do not need to consider it in the size of the formula. Therefore, the total size of formulae  $C_{x,x'}(i)$  in (8) is  $N_C = |V| |V'|^3 \lambda^4$ .

Finally, the complete set of formulae that we require to capture the existence condition of arithmetical models over discrete domains has the following total size:

$$4N_{local} + 4N_{far} + 2N_{CC} + N_C \leq 4(k + 1) |V'|^2 \lambda^2 + 4\lambda |V'|^3 (k + \lambda)^2 + 2\lambda^2 |V'|^2 (k + \lambda)^2 + |V| |V'|^3 \lambda^4.$$

### 5.3. Simplifying the existence condition

In this section, we relax the condition of the existence of an arithmetical model  $\sigma$  for sequences of symbolic valuations. First of all, we introduce an alternative condition for the existence of arithmetic models defined in [Demri and D'Souza 2007]. Before providing the definition, we introduce the notion of *order* of a path in the graph representing symbolic models. Let us consider the graph  $G_\rho$  associated with a sequence of symbolic valuations  $\rho \in SV(\phi)$  presented in Section 2.4. We consider directed paths in  $G_\rho$  whose arcs are labeled with  $<$  or  $\leq$  relations. We define *order*  $o(u)$  of a directed path  $u$  between two points  $a = (x, i)$  and  $b = (y, j)$  in  $G_\rho$  the number of  $<$ -labeled edges occurring in  $u$ . The order  $o(a, b)$  between two points is the supremum of the set  $\{o(u) : u \text{ is a directed path from } a \text{ to } b\}$ .

LEMMA 5.17 (LEMMA 6.1 IN [DEMRI AND D'SOUZA 2007]). *Let  $\rho$  be a locally consistent sequence of symbolic valuations from  $SV(\phi)$ . Then, there exists a model  $\sigma$  such that  $\sigma, 0 \models \phi$  if, and only if, the order  $o(u, v)$  of any pair of points  $u, v$  in  $G_\rho$  is finite.*

Lemma 5.17, combined with the definition of symbolic valuations as maximally consistent sets of  $\mathcal{D}$ -constraints, produces a condition for the existence of arithmetic models that is in fact too strong for our needs. Consider for example the following formula

$$\mathbf{G}(x < Xx \wedge \neg(y < Xy)) \quad (10)$$

which enforces strict increasing monotonicity for variable  $x$  and decreasing monotonicity for variable  $y$ . Figure 4 shows a symbolic model for formula (10) which does not admit arithmetic model, as  $o((x, 0), (y, 0)) = \infty$ . However, in (10)  $x$  and  $y$  are not compared, neither directly, nor indi-

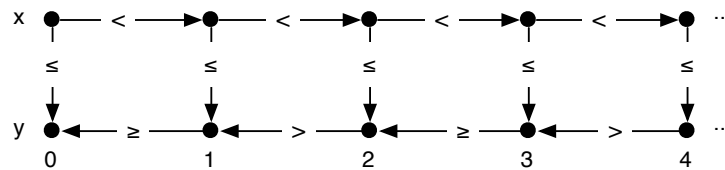


Fig. 4. Symbolic model for formula (10).

rectly, so if we disregard the relations between them in the symbolic model of Figure 4, and produce an assignment of the variables that only respects the relations between variables that are actually compared in the formula (i.e.,  $x$  with itself, and  $y$  with itself) we obtain an arithmetic model for (10). Figure 5 shows a “weaker” version of the symbolic model of Figure 4, one that only includes relations between variables compared in the formula. This weaker version of the symbolic model

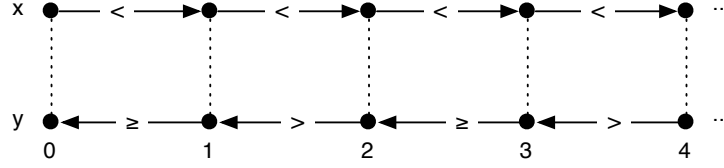


Fig. 5. A weak symbolic model for formula (10).

requires less formulae to be included in the QF-EU( $\mathcal{D}$ ) encoding than the maximally consistent one, as it does not contain any comparison between unrelated terms, hence it is more concise.

To characterize sequences of symbolic valuations which do not take into account relations among variables that are not compared with each other in a formula  $\phi$ , we first remark that  $\phi$  induces a finite partition  $\{V_1, \dots, V_h\}$  of set  $V$  such that  $x, y \in V_i$  if and only if there is a  $\mathcal{D}$ -constraint  $R(X^i x, X^j y)$  occurring in  $\phi$ , for some  $i, j \in \mathbb{Z}$ . Then, we introduce the notions of *weak symbolic valuation* and of *sequence of weak symbolic valuations* and we draw a relation with their standard definitions.

Given a symbolic valuation  $sv \in SV(\phi)$ , its *weak* version  $\bar{sv}$  is obtained by removing from  $sv$  all relations  $R(X^i x, X^j y)$  such that  $x \in V_l$  and  $y \in V_t$  implies  $l \neq t$ . We indicate with  $SV_w(\phi)$  the set of all weak symbolic valuations. Given a CLTLB( $\mathcal{D}$ ) formula  $\phi$ , its weak symbolic model  $\bar{\rho}$  is a sequence in  $SV_w(\phi)^\omega$  of weak symbolic valuations such that  $\bar{\rho}, 0 \models^{sym} \phi$ . The following proposition shows that weak symbolic models are enough to solve satisfiability problems, since they embed only information among related variables, i.e., those belonging to the same class  $V_i \subseteq V$ .

**PROPOSITION 5.18.** *Let  $\phi$  be a CLTLB( $\mathcal{D}$ ) formula,  $\rho \in SV(\phi)^\omega$ ,  $\bar{\rho}$  its weak version and  $\sigma$  a sequence of  $\mathcal{D}$ -valuations. Then,*

$$\rho, 0 \models^{sym} \phi \text{ and } \sigma, 0 \models \rho \Leftrightarrow \bar{\rho}, 0 \models^{sym} \phi \text{ and } \sigma, 0 \models \bar{\rho}.$$

**PROOF.** First of all, it is straightforward to show that, if  $v'$  is an assignment to terms (see Section 2.4), then,  $v' \models_{\mathcal{D}} f(sv)$  entails  $v' \models_{\mathcal{D}} f(\bar{sv})$ . Moreover, if  $v' \models_{\mathcal{D}} f(\bar{sv})$  then there exists a symbolic valuation  $sv'$ , such that  $sv' = \bar{sv}$  and  $v' \models_{\mathcal{D}} f(sv')$ .

Then, let  $\{V_1, \dots, V_h\}$  be the partition of set  $V$ .

The first side is immediate because all symbolic valuations in  $\rho$  contain relations between terms whose variables are not related in  $\phi$ . In particular, the symbolic valuations in  $\rho$  involve  $\mathcal{D}$ -constraints between variables belonging to different partitions of  $V$ . Therefore, from the considerations above, the weak version  $\bar{\rho}$  of  $\rho$  is still such that  $\sigma, 0 \models \bar{\rho}$ . Moreover,  $\bar{\rho}, 0 \models^{sym} \phi$  follows from the definition of  $\models^{sym}$  with respect to atomic formulae  $R(\alpha_1, \dots, \alpha_n)$ . In fact, for all  $i$  and  $R(\alpha_1, \dots, \alpha_n)$  occurring in  $\phi$ , if  $\rho, i \models^{sym} R(\alpha_1, \dots, \alpha_n)$  then  $\bar{\rho}, i \models^{sym} R(\alpha_1, \dots, \alpha_n)$ .

Conversely, suppose we have  $\bar{\rho}, 0 \models^{sym} \phi$  and  $\sigma, 0 \models \bar{\rho}$ . Then, from  $\sigma$  one can deduce from Lemma 5.3 a unique sequence of symbolic valuations  $\nu$ , by iterating the result for increasing values of  $k$ . Then, we immediately obtain that  $\sigma, 0 \models \nu$ . As discussed in the first part of the proof,  $\nu$  is a symbolic model for formula  $\phi$ . In fact, we observe that the weak version  $\bar{\nu}$  of  $\nu$  is exactly  $\bar{\rho}$ , because both are such that  $\sigma, 0 \models \bar{\rho}$  and  $\sigma, 0 \models \bar{\nu}$  (for each position they must exhibit the same relation of  $\phi$  between two (or more) terms). Moreover, we have that  $\nu, 0 \models^{sym} \phi$  because, for all  $i$  and  $R(\alpha_1, \dots, \alpha_n)$  occurring in  $\phi$ , if  $\bar{\nu}, i \models^{sym} R(\alpha_1, \dots, \alpha_n)$ , then  $\nu, i \models^{sym} R(\alpha_1, \dots, \alpha_n)$ . In other words, additional relations in symbolic valuations occurring in  $\nu$  (but not in  $\phi$ ) do not affect the truth values of the relations occurring in  $\phi$ .  $\square$

For this reason, we can use the partition of  $V$  to refine Lemma 5.17. Let  $\{V_1, \dots, V_h\}$  be a partition of  $V$  derived from  $\phi$ . Let  $\bar{\rho}$  be a locally consistent sequence of symbolic valuations from  $SV_w(\phi)$ . We say that two points  $u = (x, i)$  and  $v = (y, j) \in G_{\bar{\rho}}$  are *homogeneous* when  $x, y \in V_l$ , with  $l \in [1, h]$ .

**COROLLARY 5.19.** *Let  $\{V_1, \dots, V_h\}$  be a partition of  $V$  derived from  $\phi$ . Let  $\bar{\rho}$  be a locally consistent sequence of weak symbolic valuations from  $SV_w(\phi)$ . Then, there exists a model  $\sigma$  such that  $\sigma, 0 \models \phi$  if, and only if, the order  $o(u, v)$  of any pair of homogeneous points  $u, v$  in  $G_{\bar{\rho}}$  is finite.*

**PROOF.** The result follows from Proposition 5.18 because weak symbolic models  $\bar{\rho}$ , and consequently their graph  $G_{\bar{\rho}}$ , relate only points whose variables are homogeneous.  $\square$

## 6. COMPLETENESS

Completeness has been studied in depth for Bounded Model Checking. Given a state-transition system  $M$ , a temporal logic property  $\phi$  and a bound  $k > 0$ , BMC looks for a witness of length  $k$  for  $\neg\phi$ . If no witness exists then length  $k$  may be increased and BMC may be reapplied. In principle, the process terminates when a witness is found or when  $k$  reaches a value, the *completeness threshold* (see Definition 3.1), which guarantees that if no counterexample has been found so far, then no counterexample disproving property  $\phi$  exists in the model. For LTL it is shown that a completeness threshold always exists; [Clarke et al. 2004] shows a procedure to estimate an over-approximation of the value, by satisfying a formula representing the existence of an accepting run of the product automaton  $M \times B_{\neg\phi}$ , where  $B_{\neg\phi}$  is the Büchi automaton for  $\neg\phi$  and  $M$  is the system to be verified.

In [Bersani et al. 2011] we have already given a positive answer to the problem of whether there exists a completeness threshold for the satisfiability problem for CLTLB( $\mathcal{D}$ ), provided that  $\mathcal{D}$  satisfies suitable conditions, informally summarized here:

- ultimately periodic symbolic models of the form  $\alpha\beta^\omega$  of CLTLB( $\mathcal{D}$ ) formulae admit an arithmetic model and
- the length  $k$  characterizing the  $k$ -satisfiability tests to be checked is bounded with respect to the size of the formula.

The previous two assumptions hold when the constraint system  $\mathcal{D}$  defining atomic formulae has the completion property (see Section 2.4.1) or when condition  $C$  holds (see Proposition 2.8). In these cases, if a CLTLB( $\mathcal{D}$ ) formula  $\phi$  is satisfiable, then all ultimately periodic symbolic models  $\rho$ , such that  $\rho \models^{sym} \phi$ , admit an arithmetic model  $\sigma$  such that  $\sigma \models \rho$ . In [Bersani et al. 2011] we used an automata- and logic-based approach to show how completeness can be achieved for the satisfiability problem, where automata  $\mathcal{A}_C$  and  $\mathcal{A}_\ell$  are still explicitly constructed (see Section 2.4 for details on  $\mathcal{A}_C$  and  $\mathcal{A}_\ell$ ). In that work the CLTLB( $\mathcal{D}$ ) representations  $\phi_{\mathcal{A}_C}$  and  $\phi_{\mathcal{A}_\ell}$  of automata are used, along with the original formula  $\phi$ , to solve the satisfiability for  $\phi$ , which is reduced to a finite amount of  $k$ -satisfiability problems of formula  $\phi'$  for increasing values of  $k$ . Formula  $\phi'$  is defined as the conjunction of the original formula  $\phi$  with formula  $\phi_{\mathcal{A}_C}$  representing runs of automaton  $\mathcal{A}_C$  and formula  $\phi_{\mathcal{A}_\ell}$  representing runs of  $\mathcal{A}_\ell$ . Sequences of locally consistent symbolic valuations recognized by the automaton  $\mathcal{A}_\ell$  are, in fact, models of the formula  $\phi_{\mathcal{A}_\ell} := \mathbf{G}(\bigvee_1^m sv_i)$ . Since the bounded representation of formulae (see Section 4) is not contradictory (i.e., two consecutive symbolic valuations are satisfiable when they are locally consistent), the previous formula exactly represents words of automaton  $\mathcal{L}(\mathcal{A}_\ell)$ . Formula  $\phi_{\mathcal{A}_C}$  is derived from automaton  $\mathcal{A}_C$ , by means of the translation in [Sistla and Clarke 1985]. Automaton  $\mathcal{A}_C$  is built by complementing automaton  $\mathcal{A}_{\neg C}$  [Safrat 1988], recognizing the complement language of  $\mathcal{L}(\mathcal{A}_C)$ , which is obtained according to the procedure proposed in [Demri and D'Souza 2007]. Finally, to check the satisfiability of  $\phi$  we verify whether formula  $\phi \wedge \phi_{\mathcal{A}_C} \wedge \phi_{\mathcal{A}_\ell}$  is  $k$ -satisfiable with  $k \in \mathbb{N}$ . In order to be complete,  $k$ -satisfiability has to be checked at most a finite number of times. The existence of a finite completeness threshold is a consequence of the existence of automaton  $\mathcal{A}_\phi$  (see Section 2.4) recognizing symbolic models of  $\phi$  and lemmata 2.7 and 2.5. In fact, let  $rd(\mathcal{A}_\phi)$  be the recurrence diameter of  $\mathcal{A}_\phi$ . Then, if formula  $\phi \wedge \phi_{\mathcal{A}_C} \wedge \phi_{\mathcal{A}_\ell}$  is not  $k$ -satisfiable for all  $k \in [1, rd(\mathcal{A}_\phi) + 1]$ , then there is no ultimately periodic symbolic model  $\rho$  such that both  $\rho, 0 \models^{sym} \phi$  and there exists an arithmetic model  $\sigma$  with  $\sigma, 0 \models \rho$ . Hence, formula  $\phi$  is unsatisfiable. Otherwise, we have found an ultimately periodic symbolic model  $\rho$  of length  $k > 0$  which admits an arithmetic model  $\sigma$ . From the  $k$ -bounded solution, we have a symbolic model  $\rho = \alpha\beta^\omega$  (or  $\alpha SV(\phi)^\omega$ ) and its bounded arithmetic model  $\sigma_k$ .

The infinite model  $\sigma$  is defined from  $\sigma_k$  by iterating infinitely many times the sequence of symbolic valuations in  $\beta$ . Therefore, the completeness bound for BSP of CLTLB( $\mathcal{D}$ ) formulae is defined by the recurrence diameter of  $\mathcal{A}_\phi$ .

Thanks to the results of the previous sections, we can simplify the method presented in [Bersani et al. 2011]. We avoid the construction of automaton  $\mathcal{A}_{-C}$  through Safra's method and the construction of set  $SV(\phi)$ . In particular, we take advantage of the definition of  $k$ -bounded models of  $\phi$ . In fact, by Lemma 5.3, a finite sequence  $\sigma_k$  of  $\mathcal{D}$ -valuations induces a unique locally consistent sequence of symbolic valuations  $\rho$ , such that  $\sigma_k, i \models \rho(i)$ , for all  $i \in [0, k]$ . Therefore, formula  $\phi_{\mathcal{A}_\ell}$  is no longer needed in order to obtain a finite locally consistent sequence of symbolic valuations. If  $\phi$  is a formula of CLTLB( $\mathcal{D}$ ) and  $\mathcal{D}$  has the completion property, we can simply solve  $k$ -satisfiability problems for  $\phi$  and not for  $\phi \wedge \phi_\ell$ . When  $\mathcal{D}$  does not have the completion property, formula (6) allows us to avoid the construction of  $\mathcal{A}_C$ . In fact, by theorems 5.8 and 5.11 if  $|\phi|_k$  is satisfiable then there is an ultimately periodic run  $\alpha\beta^\omega$  which is recognized by automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$ . In case of constraint systems without completion, Theorem 5.16 guarantees that if  $|\phi|_k$  is satisfiable and formula (6) does not hold, then  $\phi$  is satisfiable. Therefore, model  $\alpha\beta^\omega$  obtained by solving the  $k$ -satisfiability problem belongs to the language recognized by automaton  $\mathcal{A}_s \times \mathcal{A}_\ell$  and also to the one of  $\mathcal{A}_C$ .

The completeness property still holds without the explicit representation of automata  $\mathcal{A}_\ell$  and  $\mathcal{A}_C$  in the formula we check for satisfiability; the completeness threshold is the recurrence diameter of  $\mathcal{A}_\phi$ . However, since we do not require  $\mathcal{A}_\phi$  to be actually built, the completeness threshold can be over-approximated as being exponential in the size of the formula: the number of control states of  $\mathcal{A}_\phi$  is  $\mathcal{O}(2^{|\phi|})$ . We can consider a rough estimation for the completeness threshold defined by the value  $|\mathcal{A}_C| \cdot |SV(\phi)| \cdot 2^{|\phi|}$ . The number of symbolic valuations  $|SV(\phi)|$  is exponential in the size of formula  $\phi$ . In case of constraint systems ( $\mathcal{D}, <, =$ ) with completion  $|\mathcal{A}_C| = 1$ . When  $\mathcal{D}$  is IPC\* control states of  $\mathcal{A}_C$  are defined by tuples of the form  $(a, i, b, j, d, h)$  where  $a, b \in V, i, j \in [0, \lambda]$ , with  $\lambda = \lceil \phi \rceil - \lfloor \phi \rfloor + 1, d, h \in \{0, 1\}$ . Then,  $|\mathcal{A}_C| = 4|V|^2|\lambda|^2$ .

## 7. RELATED WORK

Schüle and Schneider [Schüle and Schneider 2007] provide a general algorithm to decide bounded LTL( $L$ ) model-checking problems of infinite state systems where  $L$  is a general underlying logic. An LTL( $L$ ) formula  $\phi$  is translated into an equivalent Büchi automaton  $\mathcal{A}_\phi$  which is symbolically represented by means of a structure defining its transition relation and acceptance condition. Then, the LTL( $L$ ) model-checking problem is reduced to the  $\mu$ -calculus model-checking problem modulo  $L$ , i.e., a verification of a fixpoint problem for a given Kripke structure with respect to symbolic representations of  $\mathcal{A}_\phi$  and the underlying language  $L$ . Whenever properties are neither proved nor disproved over finite computations, their truth value can not be defined. For this reason, the authors adopt a three-valued logic to evaluate formulae whose components may have undefined value. Bounded model-checking is performed essentially by computing approximate fixpoint sets of the desired formula and by checking whether the initial condition is a subset of such set of states. The work of [Schüle and Schneider 2007] is based on previous results presented in [Schüle and Schneider 2004], which defines a hierarchy of Büchi automata (and, therefore, temporal formulae) for which infinite state bounded model-checking is complete. The specification language of [Schüle and Schneider 2004] is the quantifier-free fragment of Presburger LTL, LTL(PA), with past-time temporal modalities. The bounded model-checking problem is defined with respect to Kripke structures  $(S, I, R)$  and it is solved by means of a reduction to the satisfiability of Presburger formulae. In general, acceptance conditions of Büchi automata, requiring that some states are visited infinitely often, can not be handled immediately by bounded approaches which do not consider ultimately periodic models used, for instance, in the bounded model-checking approach of Biere et al. [Biere et al. 1999] or in the encoding of Büchi automata of deMoura et al. [de Moura et al. 2002]. Therefore, Schüle and Schneider follow a different approach, tailored to bounded verification, and focus on the analysis of some classes of LTL formulae, denoted  $\text{TL}_F$  and  $\text{TL}_G$ , such that the corresponding



Büchi automaton has a simpler accepting condition which does not involve infinite computations.  $TL_F$  and  $TL_G$  are the sets of LTL formulae such that each occurrence of a weak/strong temporal operator is negative/positive and positive/negative, respectively. LTL formulae are then represented symbolically by an automaton which is built using the method proposed by Clarke et al. in [Clarke et al. 1994] rather than using the Vardi-Wolper construction [Vardi and Wolper 1986].

Reducing the model-checking problem to Presburger satisfiability is a rather standard approach when dealing with infinite-state systems. Demri et al. in [Demri et al. 2010] show how to solve the LTL(PA) model-checking problem for the class of *admissible* counter systems, which are finite state automata endowed with variables over  $\mathbb{Z}$  whose transitions are labeled by Presburger formulae. In [Demri et al. 2010] the authors study the decidability of the model-checking problem for admissible counter systems with respect to the first-order CTL\* language over Presburger formulae.

Hodkinson et al. study decidable fragments of first-order temporal logic in [Hodkinson et al. 2000]. Although some axiomatizations of first-order temporal logic are known, various incompleteness results induce the authors to study useful fragments with expressiveness between that of propositional and of first-order temporal logic. Hodkinson et al. are interested in studying the satisfiability problem and they do not consider the model-checking problem, which requires a formalism defining the interpretation of first-order variables over time. In other words, variables do not vary over time and their temporal behavior is not relevant. The languages investigated by the authors are obtained by restricting both the first-order part and the temporal part.

Bultan et al. present a symbolic model checker for analyzing programs with unbounded integer domains [Bultan et al. 1999]. Programs are defined by an event-action language where atomic events are expressed by Presburger formulae over programs variables  $V$ . Semantics of programs is defined in terms of infinite transition systems where the states are determined by the values of variables. The specification language is a CTL-like temporal logic enriched with Presburger-definable constraints over  $V$ . Solving the CTL model-checking problem involves the computation of least fixpoints over sets of programs states: the abstract interpretation of Cousot and Cousot [Cousot and Cousot 1977] provides a method to compute approximation of fixpoints. Model-checking is done conservatively: the approximation technique admits false negatives, i.e., the solver may indicate that a property does not hold when it actually does. Programs are analyzed symbolically by means of symbolic execution techniques and they are represented by means of Presburger-definable transition systems where Presburger formulae represent symbolically the transition relation and the set of program states. Then, the state space is partitioned in order to reduce the complexity of verification and to obtain decidability for some classes of temporal properties, such as reachability ones. Experimental results, based on the standard Bakery algorithm and the Ticket mutual-exclusion algorithm, show the effectiveness of the method when verification involves a mutual exclusion requirement.

## 8. CONCLUDING REMARKS

The decision procedure described in this paper has been implemented in our bounded satisfiability checker *Zot*, which can be found at <http://zot.googlecode.com>. The *ae<sup>2</sup>Zot* plug-in of *Zot* solves  $k$ -satisfiability for CLTLB over Quantifier-Free Presburger arithmetic (QFP), of which IPC\* is a fragment, but it also supports the constraint system  $(\mathbb{R}, <, =)$ . Even if constraint systems like IPC\*, or fragments thereof, do not provide counting mechanism (provided, for instance, through the addition of functions like  $+$  in QFP), they can still be used to represent an abstraction of a richer transition system. In fact, functions like addition, or in general relations over counters which embed a counting mechanism, make the satisfiability problem of CLTLB undecidable (see [Demri and D'Souza 2007, Section 9.3]).

To conclude this paper, we provide two examples of use of the CLTLB(IPC\*) logic to specify and verify systems behavior, which highlight the applicability of our approach.

As a first example, we show how CLTLB over  $(D, <, =)$  can be used to specify a sorting process of a sequence of fixed length  $N$  of values in  $D$ . Let  $\mathbf{v} \in D^N$  be the (initial) vector that we want to sort and  $\mathbf{a} \in D^N$  be the vector during each step of sorting. We write  $\mathbf{v}(i)$  for the  $i$ -th component of  $\mathbf{v}$ ,  $1 \leq i \leq N$ . Notice that we will use the notation  $\mathbf{a}(i)$ , which, strictly speaking, is not a

CLTLB term; however, since the length of the array is fixed, we can use  $N$  variables  $a_i$  to represent the elements of  $\mathbf{a}$ , one for each  $\mathbf{a}(i)$ . Then, in the following, if  $\mathbf{a}(i)$  is replaced with  $a_i$ , one obtains CLTLB( $D, <, =$ ) formulae. We define a set of formulae representing a sorting process which swaps unsorted pairs of values at some nondeterministic position in the vector (we focus only on the most relevant ones). A variable  $p \in [0, N - 1]$  stores the position of elements which are a candidate pair for swapping; i.e.,  $p = i$  means that element  $\mathbf{a}(i)$  is swapped with element  $\mathbf{a}(i + 1)$ , while  $p = 0$  means that no elements are swapped (0 is not a position of the vector). A nondeterministic algorithm can swap arbitrarily two elements in  $[1, N]$ ; then, the only constraint on variable  $p$  is that it is  $0 \leq p < N$ , i.e.:  $\mathbf{G}(p < N \wedge p \geq 0)$ . An unsorted pair of values is indexed by a nonzero value of  $p$ :

$$\mathbf{G} \left( \bigwedge_{i \in [1, N-1]} p = i \Rightarrow \mathbf{a}(i) > \mathbf{a}(i + 1) \right).$$

A swap between two adjacent positions of  $\mathbf{a}$  is formalized by the following formula:

$$\mathbf{G} \left( \bigwedge_{i \in [1, N-1]} p = i \Rightarrow \mathbf{X}\mathbf{a}(i) = \mathbf{a}(i + 1) \wedge \mathbf{X}\mathbf{a}(i + 1) = \mathbf{a}(i) \right).$$

Vector  $\mathbf{a}$  is unchanged when no pairs are candidate for swapping:  $\mathbf{G}(p = 0 \Rightarrow \bigwedge_{i \in [1, N]} (\mathbf{a}(i) = \mathbf{X}\mathbf{a}(i)))$ . Through the  $ae^2Zot$  plugin of the  $\mathbb{Z}ot$  tool mentioned above we can then verify properties of the algorithm, e.g., whether there exists a way to sort array  $\mathbf{a}$  within  $k$  steps (with  $k$  the verification bound), which is formalized by the following formula:

$$\mathbf{F} \left( \bigwedge_{i \in [1, N-1]} (\mathbf{a}(i) \leq \mathbf{a}(i + 1)) \wedge \bigwedge_{i \in [1, N]} \bigvee_{j \in [1, N]} (\mathbf{a}(i) = \mathbf{v}(j)) \right).$$

A CLTLB-based approach can also be used to verify properties of Timed Automata [Alur and Dill 1994] over CLTLB specifications that directly express properties over clocks (following an approach similar to the one sketched in [Tripakis et al. 2005]). Informally, a timed automaton is a finite state automaton where transitions are labeled by atomic propositions belonging to a set  $AP$  (actions), and can have guards with conditions of the form  $x \sim c$  (where  $x$  is a clock,  $c \in \mathbb{N}$  and  $\sim \in \{<, \leq, =, \geq, >\}$ ), and clock resets of the form  $x := 0$ . Automated verification of Timed Automata is made possible considering a finite quotient of the state space of the transition system  $TS(ta)$  representing computations of a timed automaton  $ta$ . By defining a suitable equivalence relation over the set of states of  $TS(ta)$ , we can define a corresponding *region transition system*  $RTS(ta)$  with a finite set of states. States of  $RTS(ta)$  (regions) are equivalence classes of states in  $TS(ta)$  satisfying the same atomic clock constraints; essentially, they are represented by conjunctions of atomic propositions and constraints over  $(\mathbb{N}, <, =)$ . Therefore, in our approach,  $RTS(ta)$  can be translated into a CLTLB formula defining the transition relation by means of formulae of the form:  $s_i \Rightarrow \mathbf{X}s_{i+1}$  and  $s_i \Leftrightarrow \xi$  where  $\xi$  is a IPC\* formula defining the region associated with  $s_i$ .

We plan to explore these (and other) applications of the use of CLTLB in future works.

## REFERENCES

- ALUR, R. AND DILL, D. L. 1994. A theory of timed automata. *Theoretical Computer Science* 126, 2, 183–235.
- ALUR, R. AND HENZINGER, T. A. 1994. A really temporal logic. *Journal of the ACM* 41, 1, 181–204.
- BERSANI, M. M., CAVALLARO, L., FRIGERI, A., PRADELLA, M., AND ROSSI, M. 2010. SMT-based verification of LTL specification with integer constraints and its application to runtime checking of service substitutability. In *IEEE International Conference on Software Engineering and Formal Methods*. 244–254.
- BERSANI, M. M., FRIGERI, A., ROSSI, M., AND SAN PIETRO, P. 2011. Completeness of the bounded satisfiability problem for constraint LTL. In *Reachability Problems*. Lecture Notes in Computer Science Series, vol. 6945. Springer, 58–71.

- BIERE, A., CIMATTI, A., CLARKE, E., AND ZHU, Y. 1999. Symbolic model checking without BDDs. In *Tools and Algorithms for the Construction and Analysis of Systems*. Lecture Notes in Computer Science Series, vol. 1579. 193–207.
- BIERE, A., HELJANKO, K., JUNTILA, T. A., LATVALA, T., AND SCHUPPAN, V. 2006. Linear encodings of bounded LTL model checking. *Logical Methods in Computer Science* 2, 5.
- BULTAN, T., GERBER, R., AND PUGH, W. 1999. Model-checking concurrent systems with unbounded integer variables: symbolic representations, approximations, and experimental results. *ACM Transactions on Programming Languages and Systems* 21, 747–789.
- CLARKE, E., GRUMBERG, O., AND HAMAGUCHI, K. 1994. Another look at LTL model checking. In *Formal Methods in System Design*. Springer-Verlag, 415–427.
- CLARKE, E., KROENING, D., OUAKNINE, J., AND STRICHMAN, O. 2004. Completeness and complexity of bounded model checking. In *Verification, Model Checking, and Abstract Interpretation*. Lecture Notes in Computer Science Series, vol. 2937. 85–96.
- CLARKE, E., MCMILLAN, K., CAMPOS, S., AND HARTONAS-GARMHAUSEN, V. 1996. Symbolic model checking. In *Computer Aided Verification*. Lecture Notes in Computer Science Series, vol. 1102. 419–422.
- COMON, H. AND CORTIER, V. 2000. Flatness is not a weakness. In *Computer Science Logic*. Lecture Notes in Computer Science Series, vol. 1862. 262–276.
- COUSOT, P. AND COUSOT, R. 1977. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*. POPL '77. 238–252.
- DE MOURA, L. M., RUESS, H., AND SOREA, M. 2002. Lazy theorem proving for bounded model checking over infinite domains. In *Automated Deduction-CADE-18*. Lecture Notes in Computer Science Series, vol. 2392. 438–455.
- DEMRI, S. 2004. LTL over integer periodicity constraints. In *Foundations of Software Science and Computation Structures*. Lecture Notes in Computer Science Series, vol. 2987. 121–135.
- DEMRI, S. AND D'SOUZA, D. 2002. An automata-theoretic approach to constraint LTL. In *FST TCS 2002: Foundations of Software Technology and Theoretical Computer Science*. Lecture Notes in Computer Science Series, vol. 2556. 121–132.
- DEMRI, S. AND D'SOUZA, D. 2007. An automata-theoretic approach to constraint LTL. *Information and Computation* 205, 3, 380–415.
- DEMRI, S., FINKEL, A., GORANKO, V., AND VAN DRIMMELEN, G. 2010. Model-checking CTL\* over flat Presburger counter systems. *Journal of Applied Non-Classical Logics* 20, 4, 313–344.
- DEMRI, S. AND GASCON, R. 2005. Verification of qualitative  $\mathbb{Z}$  constraints. In *CONCUR 2005 - Concurrency Theory*. Lecture Notes in Computer Science Series, vol. 3653. 518–532.
- DEMRI, S. AND GASCON, R. 2006. The effects of bounding syntactic resources on Presburger LTL. Tech. Rep. LSV-06-5, LSV.
- DEMRI, S. AND GASCON, R. 2007. The effects of bounding syntactic resources on Presburger LTL. In *International Symposium on Temporal Representation and Reasoning (TIME)*. IEEE Computer Society, 94–104.
- HODKINSON, I. M., WOLTER, F., AND ZAKHARYASCHEV, M. 2000. Decidable fragment of first-order temporal logics. *Annals of Pure and Applied Logic* 106, 1–3, 85–134.
- HOLZMANN, G. 1997. The model checker SPIN. *IEEE Transactions on Software Engineering* 23, 5, 279–295.
- KAMP, J. A. W. 1968. Tense logic and the theory of linear order. Ph.D. thesis, University of California at Los Angeles.
- MICROSOFT RESEARCH. 2009. Z3: An efficient SMT solver. <http://research.microsoft.com/en-us/um/redmond/projects/z3/>.
- PRADELLA, M., MORZENTI, A., AND SAN PIETRO, P. 2012. Bounded satisfiability checking of metric temporal logic specifications. *ACM Transactions on Software Engineering and Methodology (TOSEM)*. To appear.
- SAFRA, S. 1988. On the complexity of omega -automata. In *IEEE Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 319–327.
- SCHÜLE, T. AND SCHNEIDER, K. 2004. Bounded model checking of infinite state systems: exploiting the automata hierarchy. In *Proceedings of the ACM and IEEE International Conference on Formal Methods and Models for Co-Design*. 17–26.
- SCHÜLE, T. AND SCHNEIDER, K. 2007. Bounded model checking of infinite state systems. *Formal Methods in System Design* 30, 51–81.
- SISTLA, A. P. AND CLARKE, E. M. 1985. The complexity of propositional linear temporal logics. *Journal of the ACM* 32, 3, 733–749.
- TRIPAKIS, S., YOVINE, S., AND BOUAJJANI, A. 2005. Checking timed Büchi automata emptiness efficiently. *Formal Methods In System Design* 26, 3, 267–292.
- VARDI, M. Y. AND WOLPER, P. 1986. An automata-theoretic approach to automatic program verification. In *Proceedings, Symposium on Logic in Computer Science*. IEEE Computer Society, 332–344.

## 9. APPENDIX

### 9.1. Proof of Proposition 5.4

PROOF. We show that for all  $i \geq 0$ ,  $(\pi, \sigma), i \models \phi \Leftrightarrow r_{model}(\pi, \sigma), i \models (r(\phi) \wedge \mathbf{G}(\bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0)))$ . It follows immediately  $(\pi, \sigma), i \models \phi \Leftrightarrow r_{model}(\pi, \sigma), i \models r(\phi)$  and  $r_{model}(\pi, \sigma), i \models \mathbf{G}(\bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0))$ . Hereafter, we write  $\theta$  instead of  $r_{model}(\pi, \sigma)$ .

First, we prove by induction the left subformula  $(\pi, \sigma), i \models \phi \Leftrightarrow \theta, i \models r(\phi)$  for  $i \geq 0$ . The **base case** is given on propositional atoms. Since  $(\pi, \sigma), i \models p_j \Leftrightarrow p_j \in \pi(i)$  and by definition of  $\theta = r_{model}((\pi, \sigma))$ , we can conclude that  $\theta(i, x_{p_j}) = 1$ . By definition  $\theta, i \models (x_{p_j} = 1) \Leftrightarrow \theta(i, x_{p_j}) = 1$ ; hence,  $\theta, i \models r(p_j)$ . Moreover, because  $(\pi, \sigma), i \models R(\alpha_1, \dots, \alpha_n)$  depends only on  $\sigma$  and  $\theta(j, x) = \sigma(j, x)$  for all  $x \in V$  and  $\lfloor \phi \rfloor \leq j$ , then  $\theta, i \models R(\alpha_1, \dots, \alpha_n)$ .

**Inductive step.**

- If  $\phi = \neg\psi$  then  $(\pi, \sigma), i \models \phi \Leftrightarrow (\pi, \sigma), i \not\models \psi$ . By inductive hypothesis, this is equivalent to  $\theta, i \not\models r(\psi)$ , i.e.  $\theta, i \models r(\phi)$ , as  $r(\phi) = \neg r(\psi)$ .
- If  $\phi = \psi_1 \wedge \psi_2$  then  $(\pi, \sigma), i \models \phi \Leftrightarrow (\pi, \sigma), i \models \psi_1$  and  $(\pi, \sigma), i \models \psi_2$ . By inductive hypothesis, this is equivalent to  $\theta, i \models r(\psi_1)$  and  $\theta, i \models r(\psi_2)$ , i.e.  $\theta, i \models r(\psi_1) \wedge r(\psi_2)$ , and  $\theta, i \models r(\phi)$ .
- If  $\phi = \mathbf{X}\psi$  then  $(\pi, \sigma), i \models \phi \Leftrightarrow (\pi, \sigma), i+1 \models \psi$ . By inductive hypothesis, this is equivalent to  $\theta, i+1 \models r(\psi)$ , i.e.,  $\theta, i \models \mathbf{X}r(\psi)$ , which corresponds to  $\theta, i \models r(\phi)$ .
- If  $\phi = \mathbf{Y}\psi$  then  $(\pi, \sigma), i \models \phi \Leftrightarrow (\pi, \sigma), i-1 \models \psi$  and  $i \geq 0$ . By inductive hypothesis, this is the same as  $\theta, i-1 \models r(\psi)$  and  $i \geq 0$ , i.e.,  $\theta, i \models \mathbf{Y}r(\psi)$ , and  $\theta, i \models r(\phi)$ , as  $r(\phi) = \mathbf{Y}r(\psi)$ .
- If  $\phi = \psi_1 \mathbf{U}\psi_2$  then  $(\pi, \sigma), i \models \phi \Leftrightarrow$  there exists  $j \geq i$  s.t.  $(\pi, \sigma), j \models \psi_2$  and  $(\pi, \sigma), n \models \psi_1$  for all  $i \leq n < j$ , that is, by inductive hypothesis, there exists  $j \geq i$  s.t.  $\theta, j \models r(\psi_2)$  and  $\theta, n \models r(\psi_1)$  for all  $i \leq n < j$ , which in turn is equivalent to  $\theta, i \models r(\psi_1) \mathbf{U}r(\psi_2)$  and  $\theta, i \models r(\phi)$ .
- If  $\phi = \psi_1 \mathbf{S}\psi_2$  then  $(\pi, \sigma), i \models \phi \Leftrightarrow$  there exists  $0 \leq j \leq i$  s.t.  $(\pi, \sigma), j \models \psi_2$  and  $(\pi, \sigma), n \models \psi_1$  for all  $j < n \leq i$ , that is, by inductive hypothesis there exists  $0 \leq j \leq i$  s.t.  $\theta, j \models r(\psi_2)$  and  $\theta, n \models r(\psi_1)$  for all  $j < n \leq i$ , which is equivalent to  $\theta, i \models r(\psi_1) \mathbf{S}r(\psi_2)$  and  $(\pi, \sigma), i \models r(\phi)$ .

Finally, we prove the first part  $(\pi, \sigma), 0 \models \phi \Leftrightarrow \theta, 0 \models r(\phi)$ , by taking  $i = 0$ .

Let us prove by induction the second part  $(\pi, \sigma), i \models \phi \Leftrightarrow \theta, i \models \mathbf{G}(\bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0))$ , for  $i \geq 0$ . The **base case** is  $\theta, i \models \bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0)$  which holds for all  $i \geq 0$  by definition of  $r_{model}(\pi, \sigma)$ . The **inductive hypothesis** applies on formula  $\mathbf{G}(\bigwedge_{i=1}^n x_{p_i} = 1) \vee (x_{p_i} = 0)$  at generic position  $i$  for all  $i \geq 0$ .  $\theta, i \models \bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0) \wedge \mathbf{XG}(\bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0))$  is equivalent to  $\theta, i \models (\bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0))$  and  $\theta, i \models \mathbf{XG}(\bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0))$ . The first conjunct follows from the base case. The second formula  $\theta, i \models \mathbf{XG}(\bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0))$  is equivalent to  $\theta, i+1 \models \mathbf{G}(\bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0))$  which holds by inductive hypothesis. Therefore, by taking  $i = 0$  we conclude that  $\theta, 0 \models \mathbf{G}(\bigwedge_{i=1}^n (x_{p_i} = 1) \vee (x_{p_i} = 0))$ .  $\square$

### 9.2. Proof of Proposition 5.5

PROOF. Let  $s = \lfloor \phi \rfloor$ . We show that for all  $i \geq 0$ ,  $\sigma, i \models \phi \Leftrightarrow \sigma, i+s \models p(\phi)$  by induction on the structure of the formula  $\phi$ .

The **base case** of the induction is given on the atomic formulae  $\phi = R(\alpha_1 \dots \alpha_n)$ . Since  $\sigma, i \models_{\mathcal{D}} \phi \Leftrightarrow R(\sigma(i+|\alpha_1|, x_{\alpha_1}), \dots, \sigma(i+|\alpha_n|, x_{\alpha_n}))$ , by shifting the instant  $i$  of  $s$  the satisfaction relation is  $\sigma, i \models_{\mathcal{D}} \phi \Leftrightarrow R(\sigma(i+s+|\alpha_1|-s, x_{\alpha_1}), \dots, \sigma(i+s+|\alpha_n|-s, x_{\alpha_n}))$ . Then, we can equivalently write  $\sigma, i \models_{\mathcal{D}} \phi \Leftrightarrow R(\sigma(i+s+|p(\alpha_1)|, x_{\alpha_1}), \dots, \sigma(i+s+|p(\alpha_n)|, x_{\alpha_n}))$  that is  $\sigma, i+s \models R(p(\alpha_1), \dots, p(\alpha_n))$  and  $\sigma, i+s \models p(R(\alpha_1, \dots, \alpha_n))$ . In fact, if  $\alpha = \mathbf{X}^i x$  then  $p(\alpha) = \mathbf{X}^{i-s} x$  and  $|p(\alpha)| = |\alpha| - s$ . If  $\alpha = \mathbf{Y}^i x$  then  $p(\alpha) = \mathbf{Y}^{i+s} x$  and  $|p(\alpha)| = -(i+s) = |\alpha| - s$ , since  $|\alpha| = -i$ .

**Inductive step.**

- If  $\phi = \neg\psi$  then  $\sigma, i \models \phi \Leftrightarrow \sigma, i \not\models \psi$ . By inductive hypothesis, this is equivalent to  $\sigma, i + s \not\models p(\psi)$ , i.e.  $\sigma, i + s \models p(\phi)$ , as  $p(\phi) = \neg p(\psi)$ .
- If  $\phi = \psi_1 \wedge \psi_2$  then  $\sigma, i \models \phi \Leftrightarrow \sigma, i \models \psi_1$  and  $\sigma, i \models \psi_2$ . By inductive hypothesis, this is equivalent to  $\sigma, i + s \models p(\psi_1)$  and  $\sigma, i + s \models p(\psi_2)$ , i.e.  $\sigma, i + s \models p(\psi_1) \wedge p(\psi_2)$ , and  $\sigma, i + s \models p(\phi)$ .
- If  $\phi = \mathbf{X}\psi$  then  $\sigma, i \models \phi \Leftrightarrow \sigma, i + 1 \models \psi$ . By inductive hypothesis, this is equivalent to  $\sigma, i + 1 + s \models p(\psi)$ , i.e.,  $\sigma, i + s \models \mathbf{X}p(\psi)$ , which corresponds to  $\sigma, i + s \models p(\phi)$ .
- If  $\phi = \mathbf{Y}\psi$  then  $\sigma, i \models \phi \Leftrightarrow \sigma, i - 1 \models \psi$ . By inductive hypothesis, this is the same as  $\sigma, i - 1 + s \models p(\psi)$ , i.e.,  $\sigma, i + s \models \mathbf{Y}p(\psi)$ , and  $\sigma, i + s \models p(\phi)$ , as  $p(\phi) = \mathbf{Y}p(\psi)$ .
- If  $\phi = \psi_1 \mathbf{U} \psi_2$  then  $\sigma, i \models \phi$  iff there exists  $j \geq i$  s.t.  $\sigma, j \models \psi_2$  and  $\sigma, n \models \psi_1$  for all  $i \leq n < j$ , that is, by inductive hypothesis,  $\sigma, j + s \models p(\psi_2)$  and  $\sigma, n \models p(\psi_1)$  for all  $i + s \leq n < j + s$ , which in turn is equivalent to  $\sigma, i + s \models p(\psi_1) \mathbf{U} p(\psi_2)$  and  $\sigma, i + s \models p(\phi)$ .
- If  $\phi = \psi_1 \mathbf{S} \psi_2$  then  $\sigma, i \models \phi$  iff there exists  $0 \leq j \leq i$  s.t.  $\sigma, j \models \psi_2$  and  $\sigma, n \models \psi_1$  for all  $j < n \leq i$ , that is, by inductive hypothesis  $\sigma, j + s \models p(\psi_2)$  and  $\sigma, n \models p(\psi_1)$  for all  $j + s < n \leq i + s$ , which is equivalent to  $\sigma, i + s \models p(\psi_1) \mathbf{S} p(\psi_2)$  and  $\sigma, i + s \models p(\phi)$ .

Finally,  $\sigma, 0 \models \phi \Leftrightarrow \sigma, s \models p(\phi)$  by taking  $i = 0$ .  $\square$

### 9.3. Proof of Corollary 5.7

PROOF. It is easy to see that relation  $\models^{\text{sym}}$  is not affected by rewriting  $p$ . In fact, for atomic formulae  $R(\alpha_1, \dots, \alpha_n)$  we have that  $\rho, i \models^{\text{sym}} R(\alpha_1, \dots, \alpha_n)$  if, and only if,  $p(\rho), i \models^{\text{sym}} p(R(\alpha_1, \dots, \alpha_n))$ . Let us suppose that for all assignments  $v'$ , we have  $v' \models_{\mathcal{D}} f(\rho(i))$  if, and only if,  $v' \models_{\mathcal{D}} f(R)$ . Then, since  $p$  is a rewriting of terms, the previous property is preserved provided that assignment  $f$  is replaced by a function  $f' : p(\text{terms}(\phi)) \rightarrow A$ , where  $A$  is a set of fresh variables (see Section 2.4). Standard temporal modalities are handled as for  $\models$ .  $\square$

### 9.4. Complete encoding for checking $A_C$

Local strict forward path are encoded by predicate  $f_{x,y} : \mathbb{N}^3 \rightarrow \{\text{true}, \text{false}\}$  for all pairs  $x, y \in V \cup \text{const}(\phi)$  and  $\tilde{f}$  for local forward path. Similarly, predicate  $b_{x,y} : \mathbb{N}^3 \rightarrow \{\text{true}, \text{false}\}$  for all pairs  $x, y \in V \cup \text{const}(\phi)$  and  $\tilde{b}$  for local forward path.

$f_{x,y}$	$0 \leq j \leq k$ and $h \leq m$	$0 \leq j \leq k$ and $h > m$
$f_{x,y}(j, h, m)$	$\mathbf{f}_{x,y} \Leftrightarrow \sigma_k(j + h, x) < \sigma_k(j + m, y)$	$\mathbf{f}_{x,y} \Leftrightarrow \perp$
$\tilde{f}_{x,y}(j, h, m)$	$\tilde{\mathbf{f}}_{x,y} \Leftrightarrow \sigma_k(j + h, x) \leq \sigma_k(j + m, y)$	$\tilde{\mathbf{f}}_{x,y} \Leftrightarrow \perp$

for all  $h, m \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ . When both  $x, y \in \text{const}(\phi)$  then  $f_{x,y} \Leftrightarrow x < y$  and  $\tilde{f}_{x,y} \Leftrightarrow x \leq y$  for all  $0 \leq j \leq k$  and  $h \leq m$ ;  $\mathbf{f}_{x,y} \Leftrightarrow \perp$  and  $\tilde{\mathbf{f}}_{x,y} \Leftrightarrow \perp$  for all  $0 \leq j \leq k$  and  $h > m$ .

$b_{x,y}$	$0 \leq j \leq k$ and $h \geq m$	$0 \leq j \leq k$ and $h < m$
$b_{x,y}(j, h, m)$	$\mathbf{b}_{x,y} \Leftrightarrow \sigma_k(j + h, x) < \sigma_k(j + m, y)$	$\mathbf{b}_{x,y} \Leftrightarrow \perp$
$\tilde{b}_{x,y}(j, h, m)$	$\tilde{\mathbf{b}}_{x,y} \Leftrightarrow \sigma_k(j + h, x) \leq \sigma_k(j + m, y)$	$\tilde{\mathbf{b}}_{x,y} \Leftrightarrow \perp$

for all  $h, m \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ . When both  $x, y \in \text{const}(\phi)$  then  $b_{x,y} \Leftrightarrow x < y$  and  $\tilde{b}_{x,y} \Leftrightarrow x \leq y$  for all  $0 \leq j \leq k$  and  $h \geq m$ ;  $\mathbf{b}_{x,y} \Leftrightarrow \perp$  and  $\tilde{\mathbf{b}}_{x,y} \Leftrightarrow \perp$  for all  $0 \leq j \leq k$  and  $h < m$ .

Predicates  $F$  and  $\tilde{F}$  are encoded by uninterpreted predicates  $F_{x,y} : \mathbb{N}^4 \rightarrow \{\text{true}, \text{false}\}$  and  $\tilde{F}_{x,y} : \mathbb{N}^4 \rightarrow \{\text{true}, \text{false}\}$  for all pairs of variables  $x, y \in V \cup \text{const}(\phi)$ . Backward paths are encoded by means of uninterpreted predicates  $B_{x,y} : \mathbb{N}^4 \rightarrow \{\text{true}, \text{false}\}$  and  $\tilde{B}_{x,y} : \mathbb{N}^4 \rightarrow \{\text{true}, \text{false}\}$  for all pairs of variables  $x, y \in V \cup \text{const}(\phi)$ . We use the symbol  $\mathbf{P} \in \{\mathbf{F}, \mathbf{B}\}$  in order to avoid repetition of similar formulae for predicate  $F$  and  $B$ :

$$\frac{i \in [1, k] \quad \left| \quad m \in [\lfloor \phi \rfloor, \lceil \phi \rceil] \quad \right| \quad j}{\begin{array}{l} \mathbf{P}_{x,y}(j, h, i, m) \Leftrightarrow \mathbf{P}_{x,y}(j+1, h-1, i, m) \\ \mathbf{P}_{x,y}(j, h, i, m) \Leftrightarrow \mathbf{P}_{x,y}(j-1, h+1, i, m) \end{array} \quad \begin{array}{l} h \in [\lfloor \phi \rfloor + 1, \lceil \phi \rceil] \\ h \in [\lfloor \phi \rfloor, \lceil \phi \rceil - 1] \end{array} \quad \begin{array}{l} [0, i-1] \\ [1, i] \end{array}}$$

and

$$\frac{j \in [0, k-1] \quad \left| \quad h \in [\lfloor \phi \rfloor, \lceil \phi \rceil] \quad \right| \quad i}{\begin{array}{l} \mathbf{P}_{x,y}(j, h, i, m) \Leftrightarrow \mathbf{P}_{x,y}(j, h, i+1, m-1) \\ \mathbf{P}_{x,y}(j, h, i, m) \Leftrightarrow \mathbf{P}_{x,y}(j, h, i-1, m+1) \end{array} \quad \begin{array}{l} m \in [\lfloor \phi \rfloor + 1, \lceil \phi \rceil] \\ m \in [\lfloor \phi \rfloor, \lceil \phi \rceil - 1] \end{array} \quad \begin{array}{l} i \in [j, k-1] \\ i \in [j+1, k]. \end{array}}$$

Formulae defining  $F$  and  $\tilde{F}$  are encoded:

$$\mathbf{F}_{x,y}(j, h, i, m) \Leftrightarrow \begin{cases} \bigvee_{z \in V} \bigvee_{u=\lfloor \phi \rfloor}^{\lceil \phi \rceil} \mathbf{f}_{x,z}(j, h, u) \wedge \tilde{\mathbf{F}}_{z,y}(j, u, i, m) \vee \\ \bigvee_{z \in V} \bigvee_{u=\lfloor \phi \rfloor}^{\lceil \phi \rceil} \tilde{\mathbf{f}}_{x,z}(j, h, u) \wedge \mathbf{F}_{z,y}(j, u, i, m) \end{cases}$$

$$\tilde{\mathbf{F}}_{x,y}(j, h, i, m) \Leftrightarrow \bigvee_{z \in V} \bigvee_{u=\lfloor \phi \rfloor}^{\lceil \phi \rceil} \tilde{\mathbf{f}}_{x,z}(j, h, u) \wedge \tilde{\mathbf{F}}_{z,y}(j, u, i, m)$$

for all  $j, i \in [0, k]$  with  $j < i$  and for all  $h, m \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$  such that  $j+h \leq i+m$ ,  $i+m-(j+m) > -\lfloor \phi \rfloor + \lceil \phi \rceil$ ,  $h = \lfloor \phi \rfloor$ ,  $(x = z) \Rightarrow (h \neq u)$  and for all pair  $x, y \in V \cup \text{const}(\phi)$ . When  $j = i \in [0, k]$  and  $h \leq m$ , with  $h, m \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ :

$$\begin{aligned} \mathbf{F}_{x,y}(j, h, j, m) &\Leftrightarrow \mathbf{f}_{x,y}(j, h, m) \\ \tilde{\mathbf{F}}_{x,y}(j, h, j, m) &\Leftrightarrow \tilde{\mathbf{f}}_{x,y}(j, h, m) \end{aligned}$$

When  $j + h > i + m$  then:

$$\begin{aligned} \mathbf{F}_{x,y}(j, h, j, m) &\Leftrightarrow \perp \\ \tilde{\mathbf{F}}_{x,y}(j, h, j, m) &\Leftrightarrow \perp \end{aligned}$$

Formulae defining  $B$  and  $\tilde{B}$  are:

$$\mathbf{B}_{x,y}(j, h, i, m) \Leftrightarrow \begin{cases} \bigvee_{z \in V} \bigvee_{u=\lfloor \phi \rfloor}^{\lceil \phi \rceil} \mathbf{b}_{x,z}(j, h, u) \wedge \tilde{\mathbf{B}}_{z,y}(j, u, i, m) \vee \\ \bigvee_{z \in V} \bigvee_{u=\lfloor \phi \rfloor}^{\lceil \phi \rceil} \tilde{\mathbf{b}}_{x,z}(j, h, u) \wedge \mathbf{B}_{z,y}(j, u, i, m) \end{cases}$$

$$\tilde{\mathbf{B}}_{x,y}(j, h, i, m) \Leftrightarrow \bigvee_{z \in V} \bigvee_{u=\lfloor \phi \rfloor}^{\lceil \phi \rceil} \tilde{\mathbf{b}}_{x,z}(j, h, u) \wedge \tilde{\mathbf{B}}_{z,y}(j, u, i, m)$$

for all  $j, i \in [0, k]$  with  $j > i$  and for all  $h, m \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$  such that  $j+h \geq i+m$ ,  $(j+m) - (i+m) > -\lfloor \phi \rfloor + \lceil \phi \rceil$ ,  $h = \lfloor \phi \rfloor$ ,  $(x = z) \Rightarrow (h \neq u)$  and for all pair  $x, y \in V \cup \text{const}(\phi)$ . When  $j = i \in [0, k]$  and  $h \geq m$ , with  $h, m \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ :

$$\begin{aligned} \mathbf{B}_{x,y}(j, h, j, m) &\Leftrightarrow \mathbf{b}_{x,y}(j, h, m) \\ \tilde{\mathbf{B}}_{x,y}(j, h, j, m) &\Leftrightarrow \tilde{\mathbf{b}}_{x,y}(j, h, m) \end{aligned}$$

When  $j + h < i + m$  then:

$$\begin{aligned} B_{x,y}(j, h, j, m) &\Leftrightarrow \perp \\ \tilde{B}_{x,y}(j, h, j, m) &\Leftrightarrow \perp \end{aligned}$$

Formulae capturing loops are:

$$\begin{aligned} LF_x(h) &:= F_{x,x}(\mathbf{loop} - 1, h, k, h) \\ \tilde{L}F_x(h) &:= \tilde{F}_{x,x}(\mathbf{loop} - 1, h, k, h) \\ LB_x(h) &:= B_{x,x}(k, h, \mathbf{loop} - 1, h) \\ \tilde{L}B_x(h) &:= \tilde{B}_{x,x}(k, h, \mathbf{loop} - 1, h) \end{aligned}$$

for all  $h, m \in [\lfloor \phi \rfloor, \lceil \phi \rceil]$ .

Non existence condition is  $C(x, x', i)$  as defined in Section 5.1. The existence condition of an arithmetical model is captured by the formula:

$$\bigwedge_{\substack{x, x' \in V \cup \text{const}(\phi) \\ x \neq x', x \notin \text{const}(\phi) \vee x' \notin \text{const}(\phi)}} (\mathbf{loop} - 1 \leq i_{xx'} \leq k) \wedge \neg C_{x,x'}(i_{xx'}). \quad (11)$$